

IM500041

## Punches, Embosses, and Patterns, Oh My! Creative iFeatures and Punches

Pete Strycharske  
D3 Technologies

### Learning Objectives

- Learn about important design considerations and characteristics of a successful iFeature
- Create a compound iFeature that will affect a component in multiple directions
- Create sheet metal emboss features
- Implement internal patterns in iFeatures and tie iFeature values to host component parameters

### Description

When it comes to repeatable computer numerical control (CNC) patterns, sheet metal cut features, and various other Inventor part features, it's tough to beat a well-crafted iFeature. In this course, we'll cover some of the lessons the speaker developed from years of working with various clients and their requirements for the repeatable features, such as: how to create compound features (features in two directions), how to place standard iFeatures using a single point, how to control the input values for the feature, etc. We'll also explore some of the more interesting iFeatures and sheet metal punches that the speaker has built, including sheet metal embosses, dimples, and more, as well as pattern iFeatures that can be driven using the host part parameters. So, whether you're a novice user or a wily iFeature veteran, there should be a little something for everybody.

# AUTODESK UNIVERSITY

## Speaker

I am an implementation consultant with D3 Technologies, a Platinum Autodesk Partner and Authorized Training Center, based out of our Minneapolis office. I focus primarily on the following areas: engineering design and manufacturability, design automation and configuration, process efficiency and manufacturing layouts. Typically, I will partner with clients to perform an assessment of a design or process, determine some improvements, propose a path forward and develop content / mentor users to implement the project. With D3 partnering with MG-AEC, I'm really excited to explore ways to make Industrialized Construction more of a reality and am thankful to work with an organization serious about getting things done. There are so many ways to improve efficiency and profitability in the construction industry, and I'm eager to get started with that, but even more than that, I want us to be better stewards of God's good Earth. We're currently working on ways to effectively link Revit with Inventor to smoothly transition from architectural to fully manufacturable models. I'm also an Autodesk Certified Instructor and professionally certified in AutoCAD, Inventor Professional and Fusion 360. I frequent the Inventor and Factory Design Forums / Idea Stations, so if you ever have a question, please just ask! This is the 4th year that I have been blessed to speak at AU and I'm eager to share what I know with you and looking forward to soaking up information from all the other fantastic sessions. When I'm not CAD'ing things up, I love serving at my church, particularly working with the youth; playing basketball as much as my body lets me and biking. My family and I enjoy playing games together, I'm quite a slouch at Minecraft, and exploring our beautiful world. We're always looking for recommendations on the next adventure!



# AUTODESK UNIVERSITY

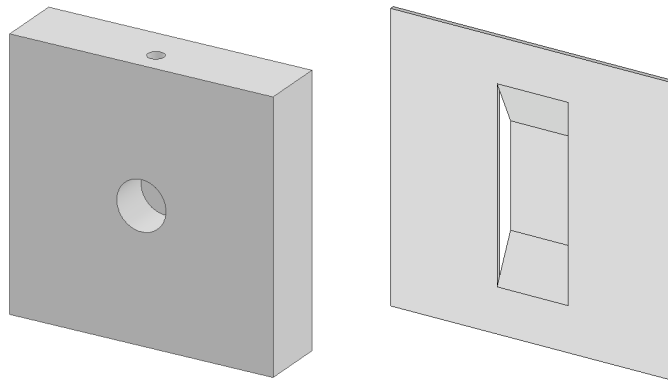
## Table of Contents

<a href="#">The Goal</a>	4
<a href="#">Why are iFeatures important?</a>	4
<a href="#">What is going to be covered?</a>	4
<a href="#">Important Considerations for iFeatures</a>	5
<a href="#">Set up a library of iFeature base models</a>	5
<a href="#">Utilizing parameters effectively</a>	6
<a href="#">Minimizing reference geometry</a>	7
<a href="#">Simplifying input geometry</a>	8
<a href="#">iFeature point placement example</a>	9
<a href="#">Bonus Tip: Naming an iFeature</a>	11
<a href="#">Create Compound iFeatures</a>	13
<a href="#">What is a compound iFeature</a>	13
<a href="#">Working from a single design sketch</a>	13
<a href="#">Multiple feature generation</a>	16
<a href="#">Create Embossed Sheet Metal Punches</a>	20
<a href="#">Traditional emboss punches</a>	20
<a href="#">Traditional emboss example</a>	21
<a href="#">Open ended formed punches</a>	25
<a href="#">Open ended emboss example #1</a>	25
<a href="#">Open ended emboss example #2</a>	29
<a href="#">A word on utilizing fillets with emboss punches</a>	39
<a href="#">Internal Patterns in iFeatures</a>	40
<a href="#">Setting up the patterned iFeature</a>	40
<a href="#">Placing a patterned iFeature</a>	44
<a href="#">Linking the pattern iFeature to host component parameters</a>	45
<a href="#">Conclusion</a>	49
<a href="#">Acknowledgements</a>	50
<a href="#">Appendix A: Weblinks to Video Demonstrations</a>	51
<a href="#">Appendix B: iLogic Examples</a>	52
<a href="#">Appendix C: iFeature Naming Considerations</a>	53

## The Goal

### Why are iFeatures important?

If you find yourself repeatedly modeling the same features or variations of features in multiple components, then an iFeature may be a good fit for your designs. In general, an iFeature can be considered as a repeatable operation, such as a sheet metal punch or a CNC pattern. iFeatures can be a singular configuration of values or can vary to offer designers maximal flexibility.



*EXAMPLE IFEATURES: COMPOUND, EMBOSS AND INTERNAL PATTERN*

### What is going to be covered?

We'll be looking at a few important topics to consider when creating iFeatures and explore some examples of "exotic" iFeatures that I've generated or seen over the years.

Important topics include:

- Utilizing parameters effectively
- Minimizing reference geometry
- Simplifying input geometry

iFeature examples include:

- Point placement iFeatures (not Sheet Metal Punches)
- Compound iFeatures
- Embossed or formed Sheet Metal features (not just cuts)
- Internal pattern iFeatures (and tying parameters to the host component)

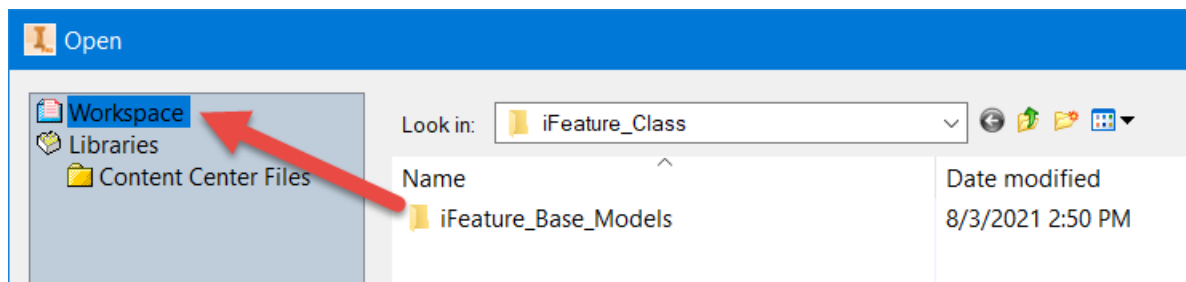
## Important Considerations for iFeatures

iFeatures can be authored in a number of ways, but if one takes a methodical approach, these features can be generated more efficiently and, even more importantly, can be placed and used more easily by the end users. There are, of course, many different considerations that go into constructing an iFeature, but we'll be covering a small subset that I've found to be useful with most of the iFeatures I've created.

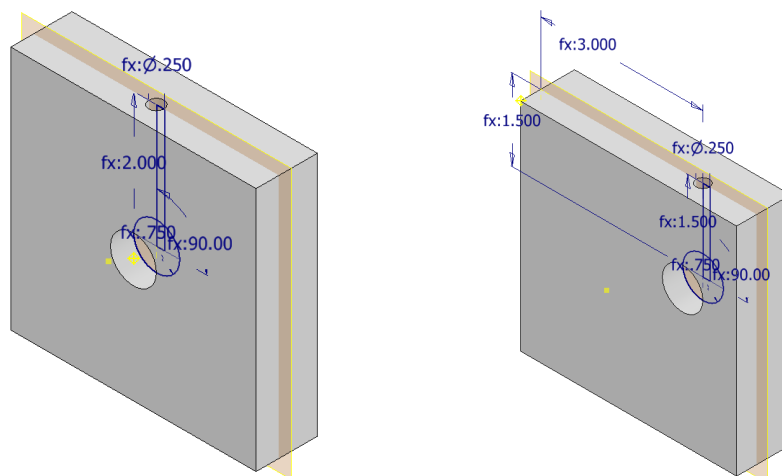
### Set up a library of iFeature base models

This may not seem obvious, but one can generate an iFeature from a model that is NEVER saved. I feel this is a bad practice. One should create a location within the Inventor project file to store all models that serve as the basis for generating iFeatures. This offers a couple of advantages.

1. This probably won't happen to anyone reading this, but sometimes I make mistakes. If these mistakes are too time consuming or not possible to fix by editing the published iFeature (the .IDE file), it is really handy to have that base model to modify and publish from.
2. There are times when a specific or one-off version is desired and the base model serves as the perfect vehicle to make the desired changes, or create a copy to make the changes, when publishing this version.



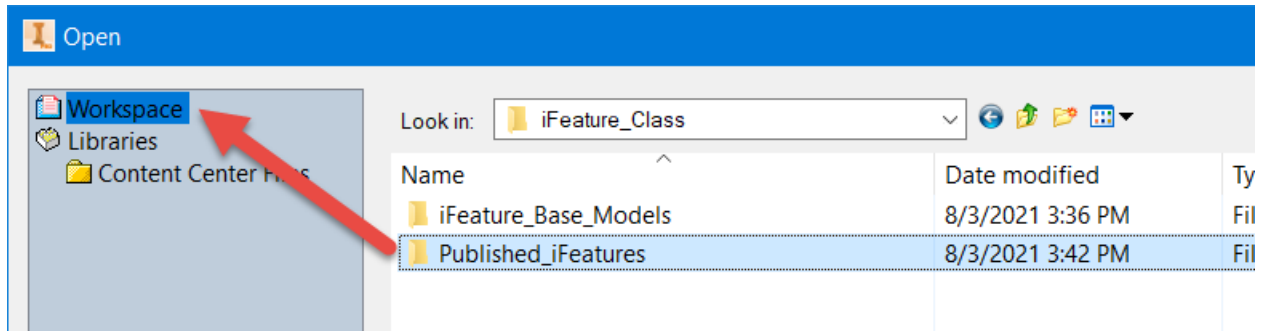
*CREATE AND UTILIZE A WORKSPACE FOLDER TO STORE THE MODELS USED TO GENERATE IFEATURES*



*INITIAL IFEATURE AT THE LEFT AND A CORNER LOCATION IFEATURE COPY AT THE RIGHT*

# AUTODESK UNIVERSITY

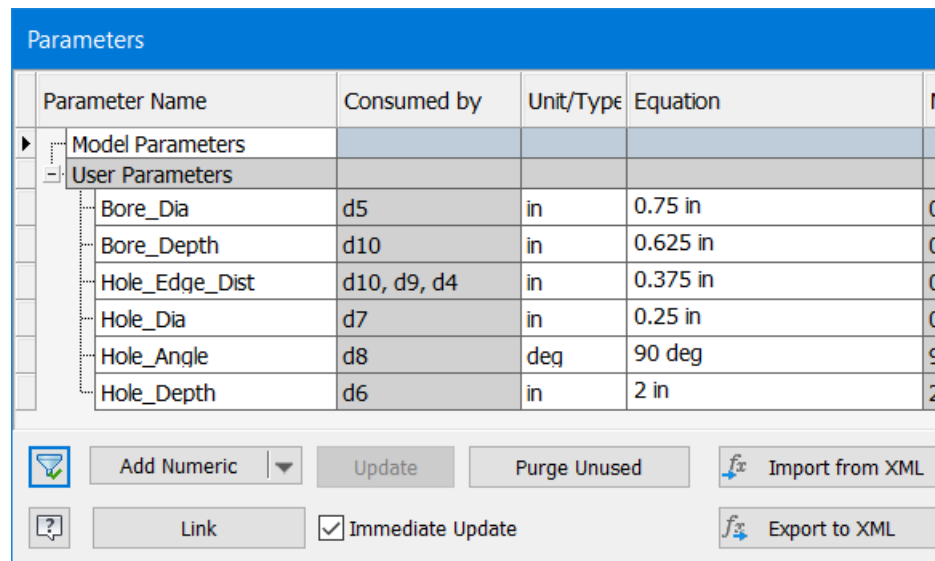
An additional tip is to set up a folder within the Inventor workspace where all the iFeatures and Sheet Metal Punches are published. This will ensure that everyone is using the same iFeatures, make them easier to access and eliminate any file location warning messages.



UTILIZE A WORKSPACE FOLDER WHERE ALL IFEATURES AND PUNCHES ARE PUBLISHED

## Utilizing parameters effectively

Parameters are the backbone of almost every Inventor design and iFeatures are no different. To ensure that parameters are utilized effectively, I recommend taking time to consider all the ways the iFeature is going to be used and then create User Parameters to meet all the requirements. Additionally, I recommend developing a consistent naming convention, which will make modifying iFeatures easier and improve the linking of iFeature and host component parameters during placement.



THINK THROUGH THE IFEATURE FUNCTIONALITY AND DEVELOP USER PARAMETERS ACCORDINGLY

Parameters					
Parameter Name	Consumed by	Unit/Type	Equation	Not	
Model Parameters					
User Parameters					
OA_Length	d0, Hole_Spacing	in	18 in	18.	
End_Dist	d4, Hole_Spacing	in	3 in	3.0	
Hole_Spacing	d14	in	$(OA\_Length - End\_Dist * 2) / (Hole\_Qty - 1)$	4.0	
Hole_Qty	d12, Hole_Spac...	ul	4 ul	4.0	

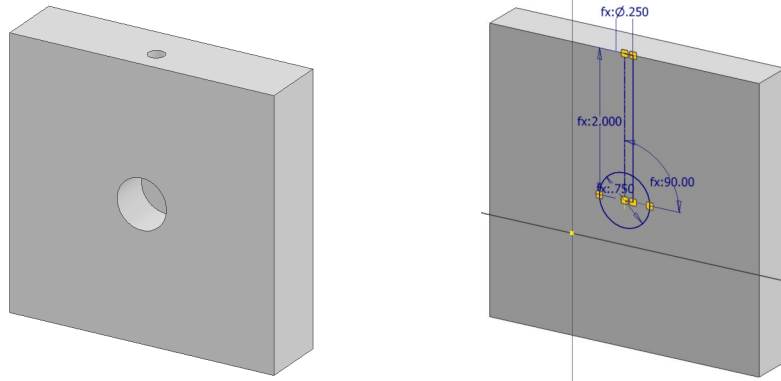
Add Numeric ▼
 Update
 Purge Unused
 Import from XML
 Link
 ☒ Immediate Update
 Export to XML

CAREFULLY CONSIDER THE NAMING CONVENTIONS FOR CONSISTENCY AND IMPROVE DOWNSTREAM USE

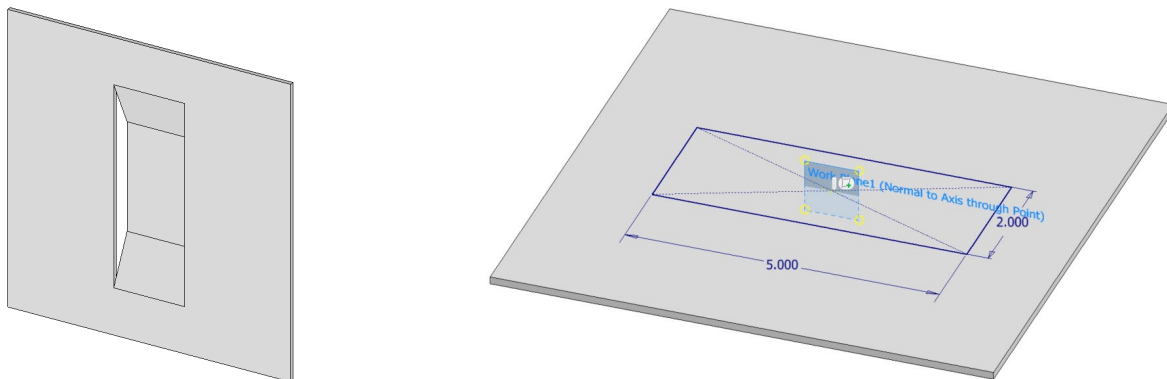
## Minimizing reference geometry

It is WAY too easy to create iFeatures that require multiple references for accurate placement. When building iFeatures consider the end users and how many reference selections will be required. If there are more than 2 or 3, one runs the risk that end users will make incorrect selections or in some cases will be unsure what to select. To avoid this, I recommend the following:

1. Select a singular geometric reference to position all features. This is usually some kind of work point and often is used for positioning the placed iFeature. We'll explore this topic with more detail in the next section.
2. Try to drive all features from a single design sketch. This may involve some thought and creativity, but if possible, driving all features from a single sketch will eliminate the need for additional reference selections.
3. If multiple design sketches are required, base each new sketch on the initial geometry or sketch. By basing new sketches off the original sketch, placement geometry will be satisfied by locating that original sketch. This includes the generation of work planes, for multi-directional iFeatures. Such work planes should be generated from the sketch geometry, not the base model solid body geometry.



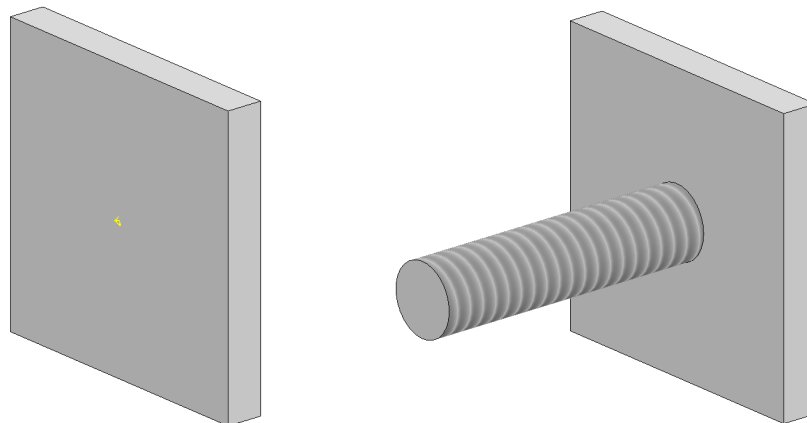
*MULTI-DIRECTIONAL IFEATURES BEING DRIVEN FROM A SINGLE DESIGN SKETCH*



*MULTI-DIRECTIONAL IFEATURES BEING DRIVEN FROM SECONDARY SKETCHES DERIVED FROM THE ORIGINAL SKETCH*

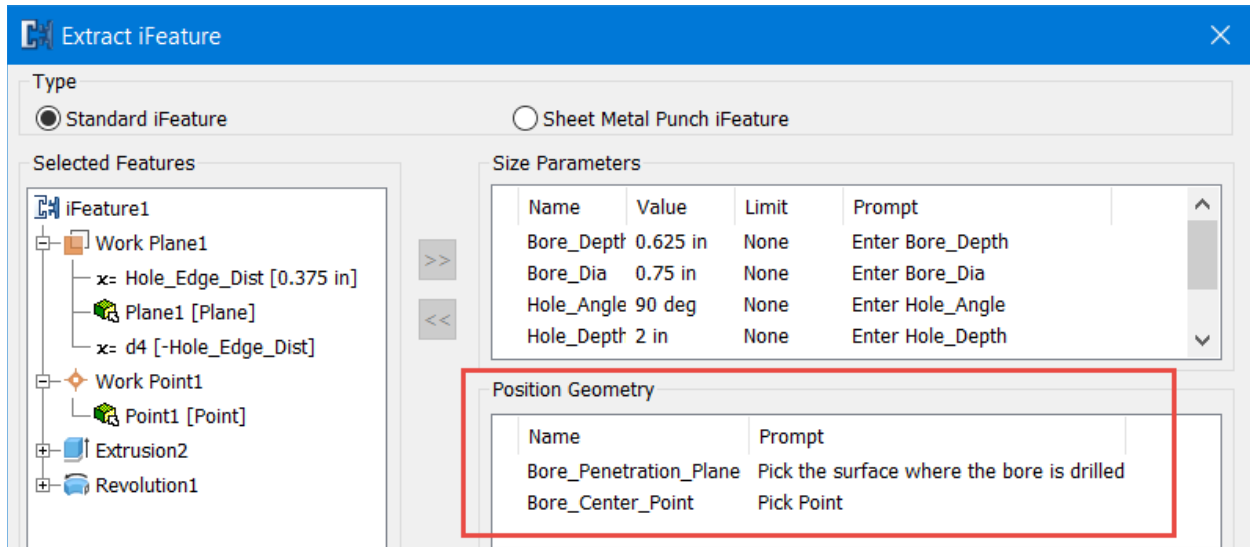
## Simplifying input geometry

Directly related to the topic above is the idea that we wish to reduce the number of selections that an end-user needs to make to accomplish a goal. Techniques such as basing a series of features on a single work point or renaming the placement geometry tags will ensure correct and efficient iFeature placement.



*A SIMPLE PEG IFEATURE LOCATED BY A SINGLE WORK POINT*



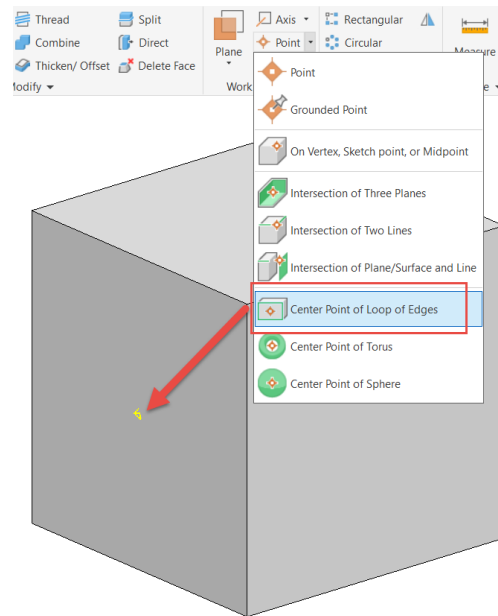


*RENAMING PLACEMENT GEOMETRY CAN HELP IMPROVE THE PLACEMENT EXPERIENCE*

## iFeature point placement example

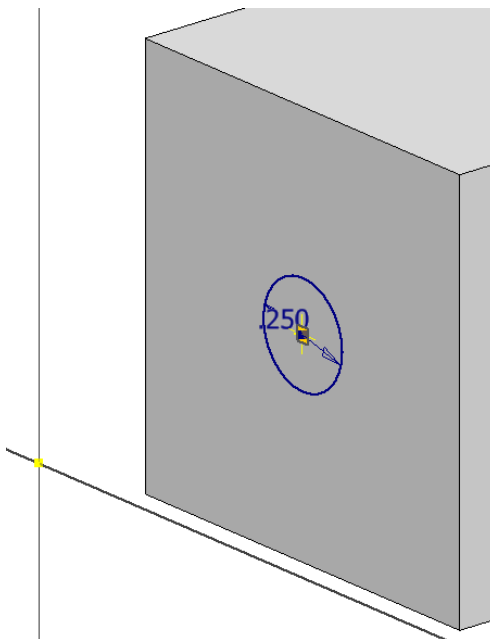
Let's take a look at the point placement with the first iFeature example of this paper. This will also give a preview of how tutorials are laid out in print. Please note that you can view [a video](#) of this process in Appendix A.

Step 1. Set up the geometry and place a work point at the center of one of the faces. It is generally recommended to not use the Origin center point as a reference, to maximize placement flexibility. Some users I respect have reported that tying iFeatures to the origin geometry will force placement at the origin, which may not always be the intention.



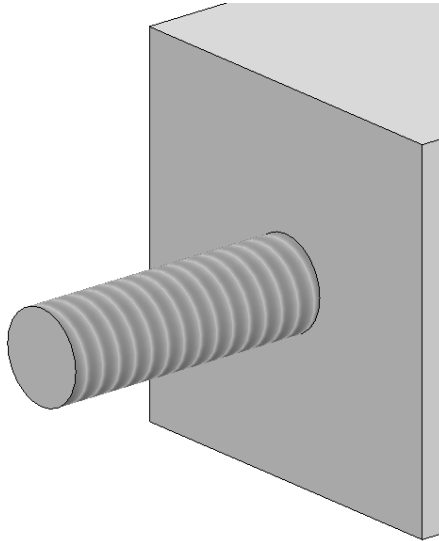
*CREATE THE BASE MODEL GEOMETRY AND PROJECT A CENTERED WORK POINT ON A SURFACE*

Step 2. Create a sketch on the surface that has the work point, project the work point onto the sketch and draw a center point circle on that projected point. Of course, any diameter will work, but one must use a diameter close to the thread size that is desired. In this example, I'll use 0.250 in as I'm planning on using 1/4-20 threading.



*GENERATE A NEW SKETCH, PROJECT THE WORK POINT AND SKETCH A CIRCLE AT THAT POINT*

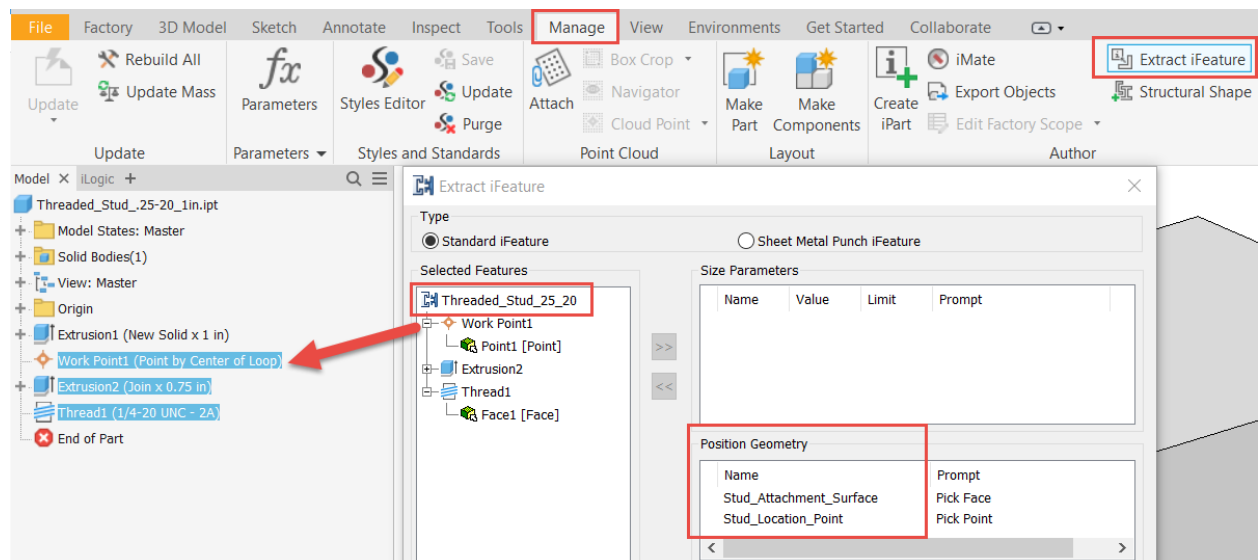
Step 3. Extrude the circle to the desired length, I'll be using 0.75 in. Place a Thread feature on the entire circle and accept the default sizing of 1/4-20.



*EXTRUDE THE CIRCLE AND APPLY A THREAD FEATURE TO THE CYLINDRICAL SURFACE*

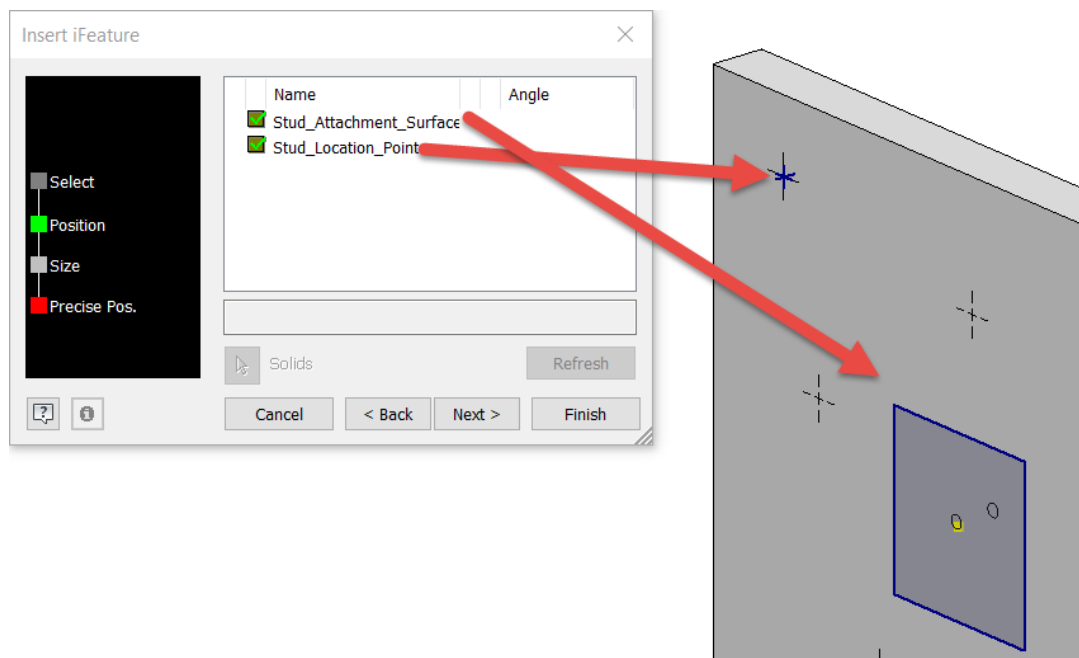
Step 4. Extract the iFeature and configure the dialog box as shown below. Be sure to select the work point, as well as the extrusion and thread features to aid in the placement. By clicking inside the of the appropriate field, we can edit the names of the placement geometry, as shown below.

**Bonus Tip:** You can rename iFeatures by right clicking on the iFeature name in the upper-right corner of the dialog box and choosing the Rename option. Be sure to avoid special characters and only use underscores instead of dashes. For a list of characters to avoid, please [see Appendix C](#).

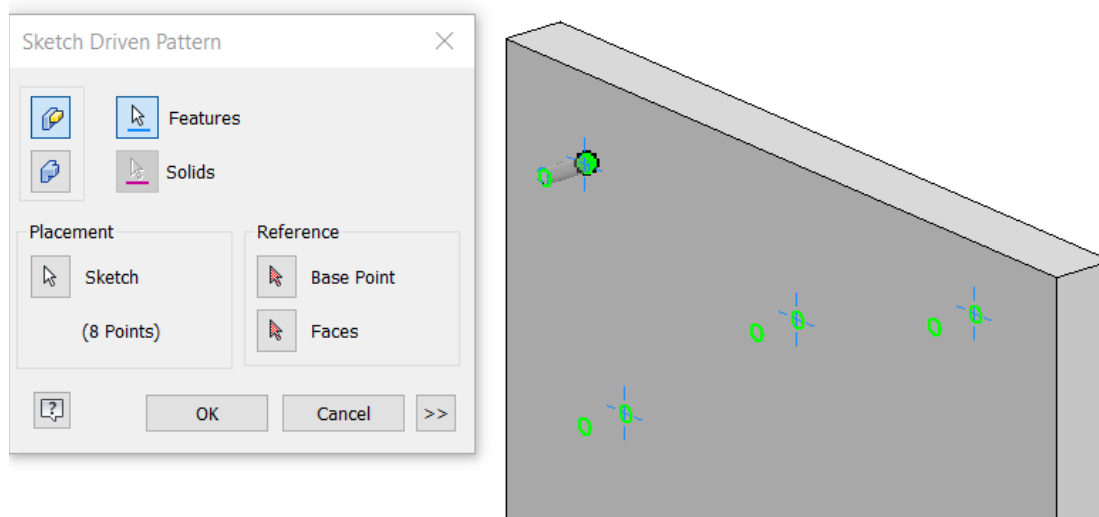
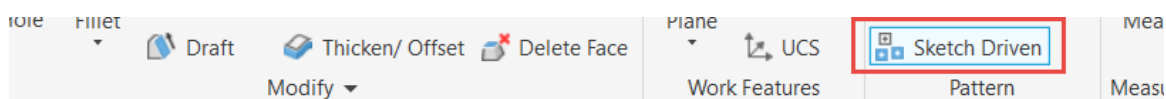


*EXTRACT THE IFEATURE BY CHOOSING THE PROPER FEATURES AND RENAMING THE FEATURE AND POSITION GEOMETRY AS APPROPRIATE*

Step 5. **ALWAYS** test out your iFeature before publishing for team-wide consumption, to work out the kinks and avoid any costly errors. To test out this iFeature, simply create a plate and a sketch with lots of center points. (Optional) utilize a sketch-driven pattern to control the location and quantity of the threaded stud iFeature.



*GENERATE A NEW PLATE COMPONENT WITH SKETCHED CENTER POINTS AND INSERT THE iFEATURE AT ONE OF THE POINTS. NOTICE THE USEFUL NAMING OF THE PLACEMENT GEOMETRY*



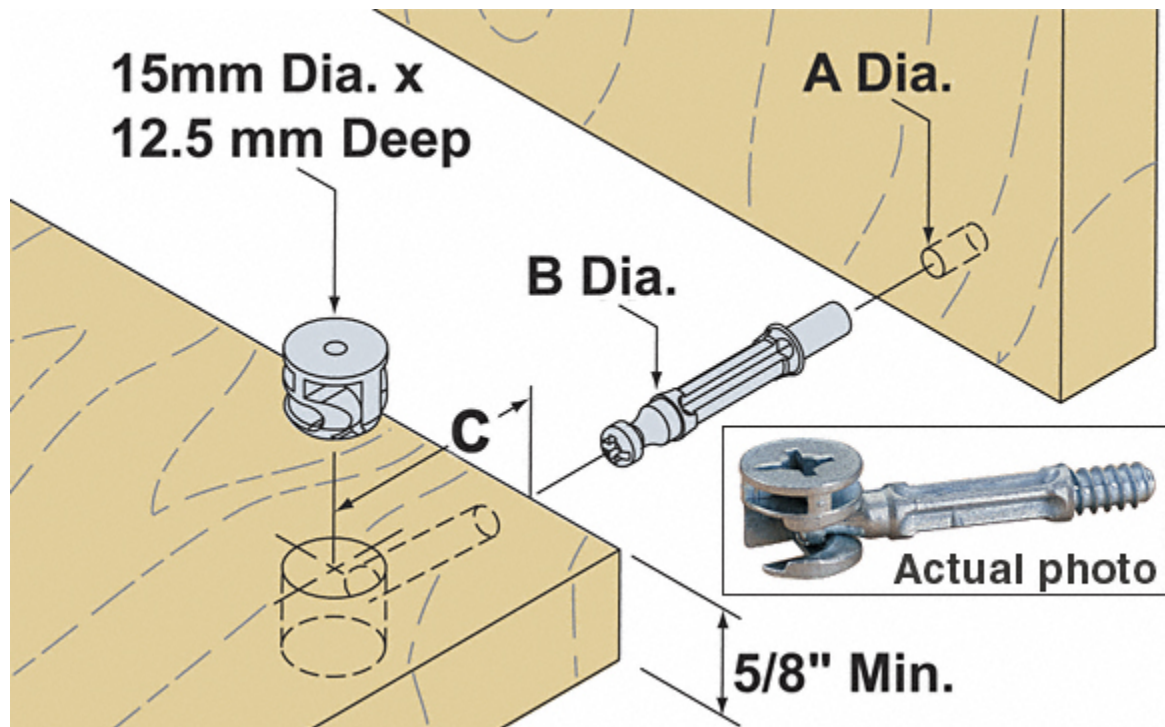
*(OPTIONAL) UTILIZE A SKETCH DRIVEN FEATURE PATTERN TO PLACE ADDITIONAL COPIES OF THE THREADED STUD*

## Create Compound iFeatures

Now that we've covered some important considerations when authoring iFeatures, let's dive into some specific examples.

### What is a compound iFeature

You may be asking yourself "what the heck is a compound iFeature (and hopefully "how do I build one)?" When I use the term "Compound iFeature" I simply mean that the iFeature affects the component in multiple directions, simultaneously. This can be used in TONS of different ways, but for this example, I'm going to use something that is hopefully familiar to just about everyone, an IKEA cam nut. Well specifically, I'm going to create the bore that would accept the cam nut, as well as the fastener the nut cams against. For your reference, please see below.



EXAMPLE IKEA FEATURE SHOWING THE CAM NUT BORE AND SCREW HOLE AT THE BOTTOM LEFT

### Working from a single design sketch

To ensure the efficient placement of these types of iFeatures, it is always best to try and drive all features from a single design sketch. While this isn't always possible, or practical, I spend some time trying to determine if there's any way to make it work. This may involve some creativity, but in this example, we can definitely drive both features from the same sketch.

To see the complete process for building this iFeature, please see [the video link](#) in Appendix A.

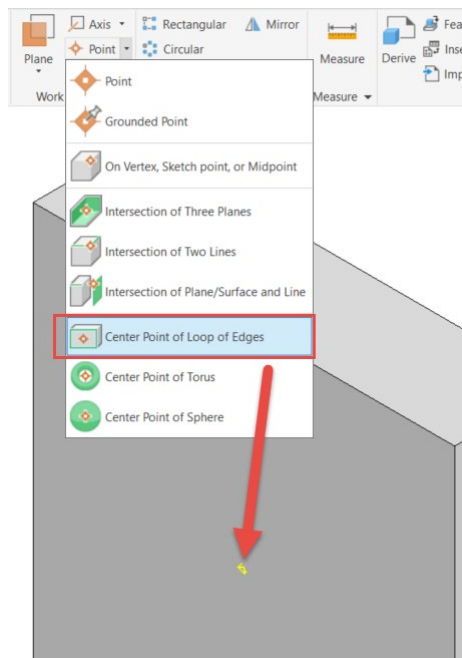
Step 1. We need to define all the relevant parameters prior to creating any of the geometry. This ensures that we've thought through the feature design sufficiently and can help determine if a single sketch is possible.

Parameters					
Parameter Name	Consumed by	Unit/Type	Equation		N
Model Parameters					
User Parameters					
Bore_Dia	d5	in	0.75 in		0
Bore_Depth	d10	in	0.625 in		0
Hole_Edge_Dist	d10, d9, d4	in	0.375 in		0
Hole_Dia	d7	in	0.25 in		0
Hole_Angle	d8	deg	90 deg		9
Hole_Depth	d6	in	2 in		2

Add Numeric
 Update
 Purge Unused
 Import from XML
 Link
 ☒ Immediate Update
 Export to XML

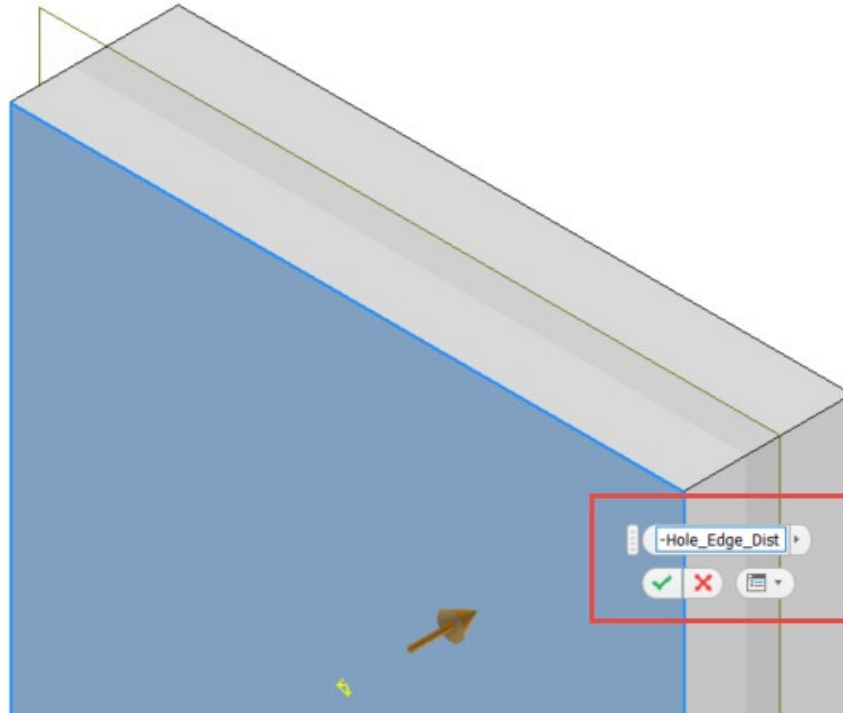
SET UP THE PARAMETERS PRIOR TO MODELING THE GEOMETRY

Step 2. Create the geometry off-center from the origin geometry and project a work point at the center of a loop of edges.



CREATE THE BASE MODEL GEOMETRY AND A WORK POINT AT THE CENTER OF ONE GROUP OF EDGES

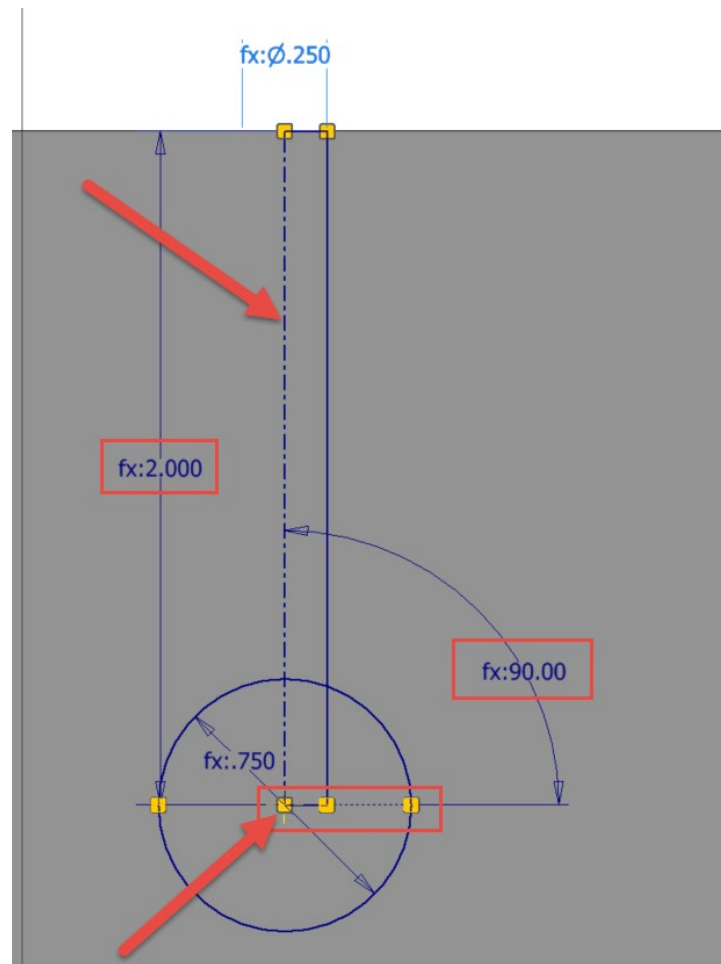
Step 3. Offset a work plane from the front surface using the distance of the hole from that edge of the part. This is the first creative step in the process. By utilizing a plane inside the surface that aligns with the center of the screw hole, we give ourselves the possibility of driving the bore and hole features from the same sketch.



*CREATE AN OFFSET WORK PLANE FROM THE FRONT SURFACE USING THE HOLE EDGE DISTANCE PARAMETER*

Step 4. Create the design sketch on the new work plane and utilize sketch, constraint and dimension tools to build the geometry required to complete both the bore and the screw hole. Be sure to note the following important items.

- Locate the center of the bore on the projected work point. This will also serve at the location of the placed iFeature.
- Locate the centerline of the screw hole at the center of the bore circle. This will ensure that the hole is centered on the bore.
- Convert that centerline to the centerline format. This helps keep the sketch organized for editing later, more easily allows for linear diameter dimensions and clarifies the Revolve Feature.
- Utilize the "Hole\_Depth" parameter, NOT a projected edge to control the depth of the screw hole.
- To control the angle of the screw hole either, draw the rectangle at an angle or remove the vertical constraint. Be sure to create a construction line between the center of the bore circle and one of its horizontal quadrant points. Then you can use the "Hole\_Angle" constraint to drill the hole at any angle, in case you desire to model the screw hole out of the side of a board vs. the top.



DESIGN SKETCH WITH IMPORTANT CHARACTERISTICS HIGHLIGHTED

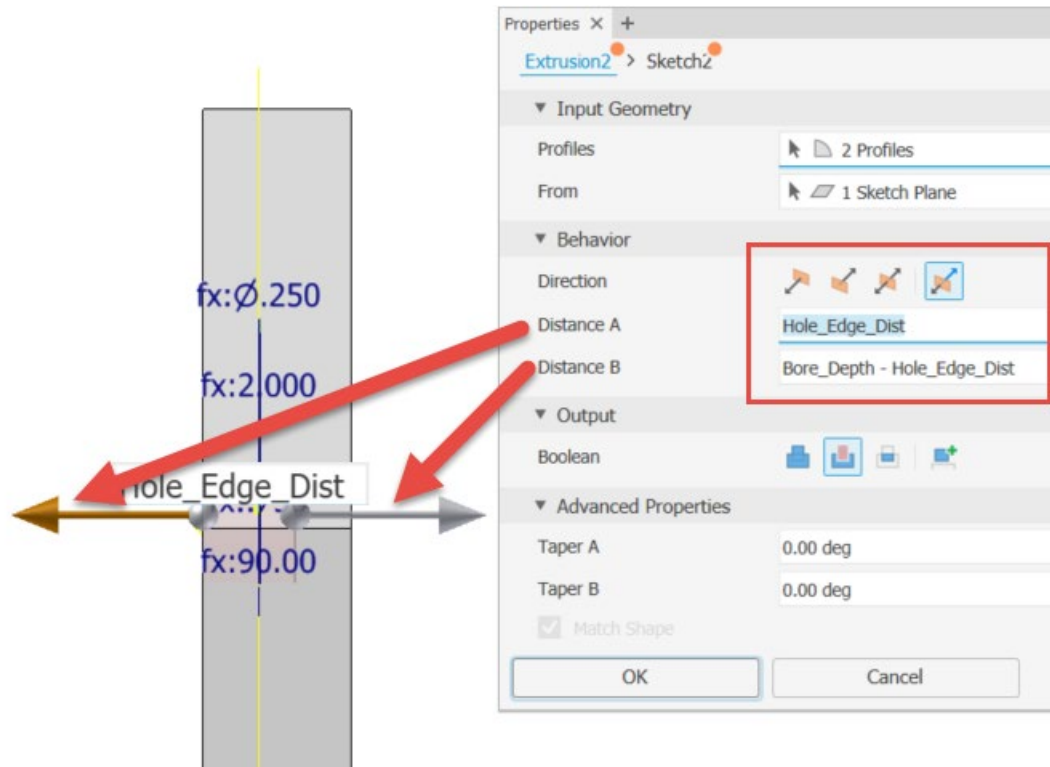
## Multiple feature generation

Once the sketch is fully constrained, we're ready to move on to creating the individual features.

Step 5. Use the Extrude command to create the bore feature. Since the sketch is buried inside the solid, we'll have to get creative a second time and utilize the asymmetric distance option of the command.

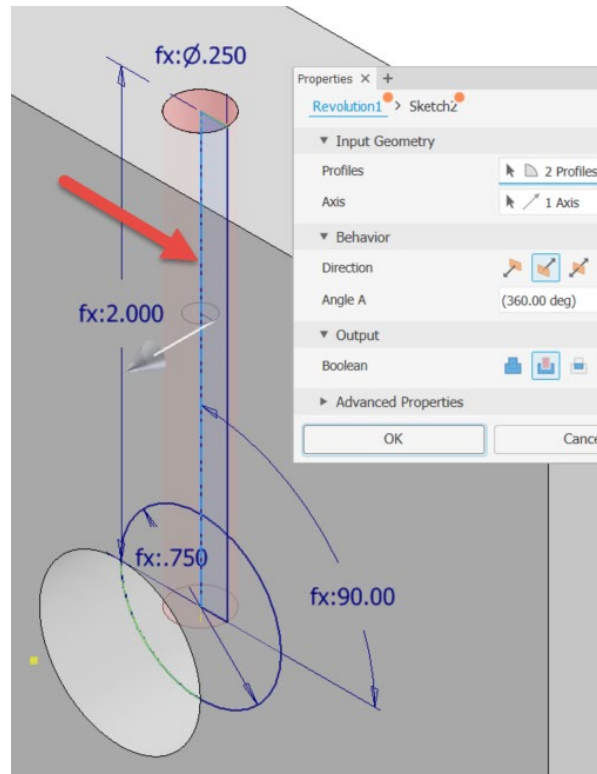
- To extrude back to the front face, use the "Hole\_Edge\_Dist" parameter for the distance in that direction.
- To extrude to the bottom of the bore, we'll have to perform some dreaded math. Thankfully it's not too difficult and we'll utilize the "Bore\_Depth" – "Hole\_Edge\_Dist" to drill down to bottom of the bore.





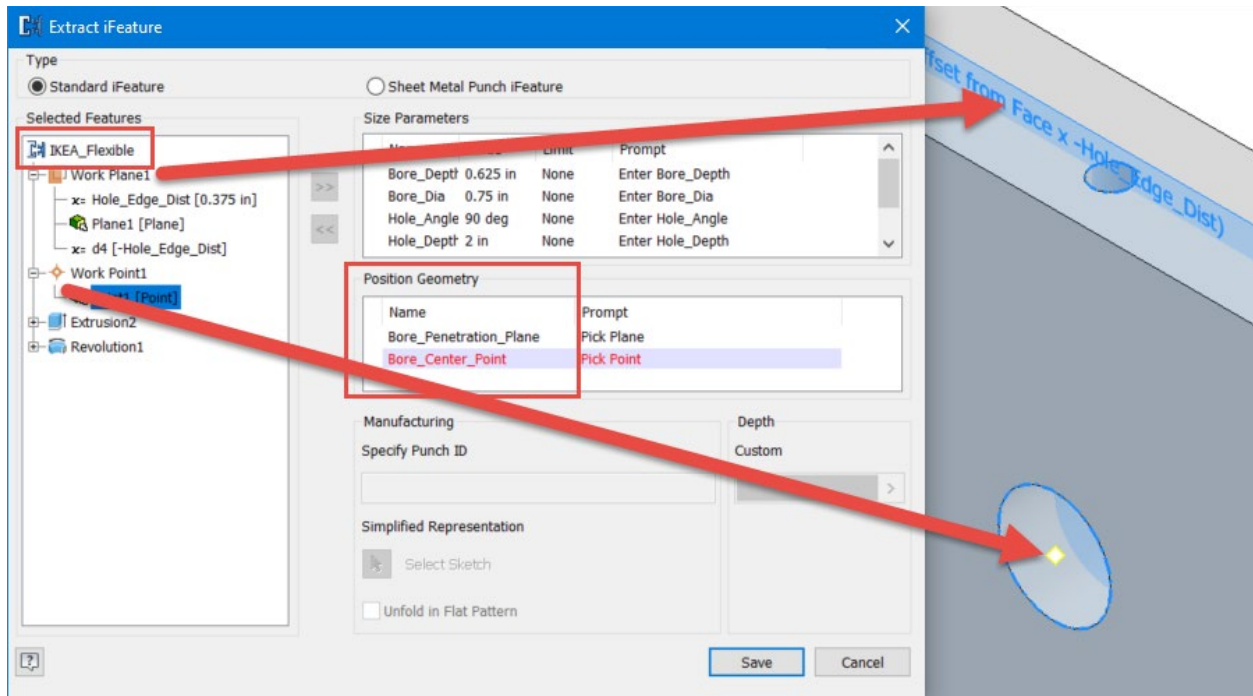
ASYMMETRIC EXTRUDE SETTINGS FOR THE BORE FEATURE

Step 6. Share the design sketch so that it can be used to drive the Revolve feature. Utilize the centerline as the central axis and perform a full revolve cut.



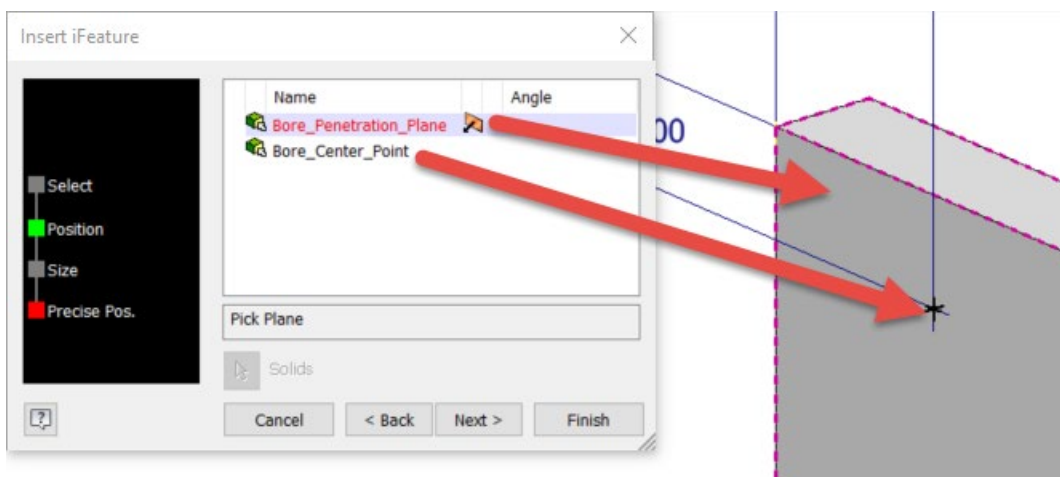
*REVOLVE CUT THE RECTANGLE FROM THE SHARED SKETCH*

Step 7. Publish the iFeature. Select the work point and work plane, as this will properly locate the feature to the penetration surface and sketch points placed in the host components. Be sure to rename the iFeature itself as well as any placement geometry.

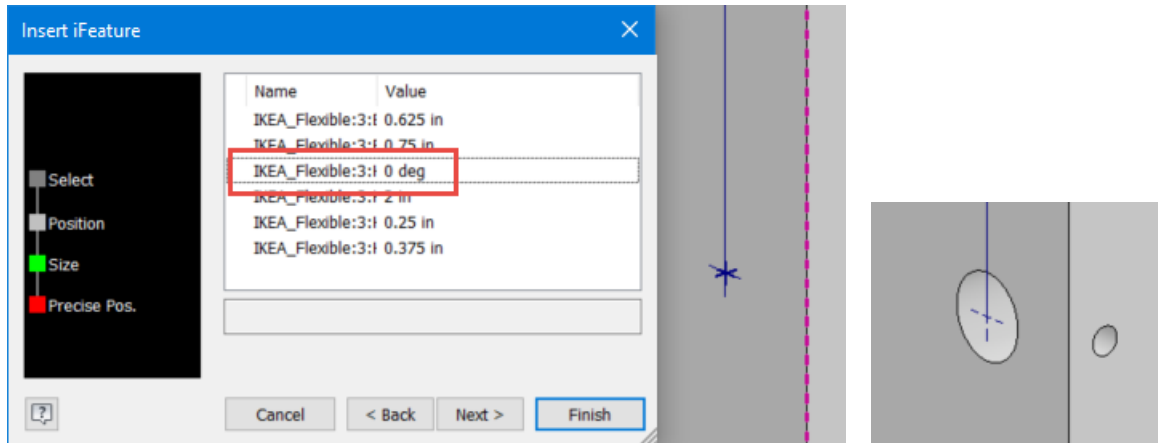


*EXTRACT THE IFEATURE BY SELECTING THE APPROPRIATE FEATURES AND RENAMING THE IFEATURE AND PLACEMENT GEOMETRY AS DESIRED*

Step 8. Test the iFeature by placing it into a host component. Create a new part and a sketch with center points, positioned where the iFeature is to be located. Edit the iFeature and readjust values as desired.



*PLACE THE IFEATURE USING THE FRONT SURFACE OF THE PART AND ONE OF THE SKETCHED CENTER POINTS*



*EDIT THE IFEATURE IF PLACED NEAR AN EDGE NOT NEAR THE TOP AND ADJUST THE ANGLE, IN THIS CASE THE ANGLE TO 0 DEGREES, AND NOTE THE SCREW HOLE WILL PASS THROUGH A SIDE FACE*

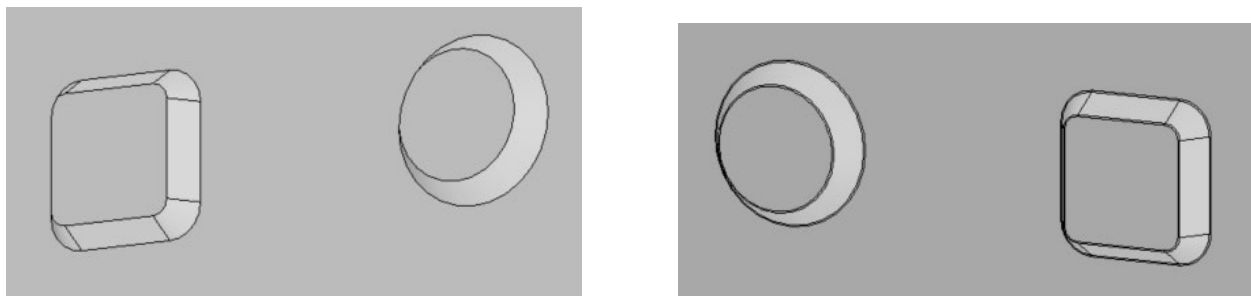
## Create Embossed Sheet Metal Punches

Most Sheet Metal punches consist of solely Cut features and that is indeed what many punches actually do in industry. However, Inventor comes with some out-of-the-box punch tools that are more along the lines of a forming operation and there are lots of times that we'll want to represent our own unique formed features. While this can be accomplished with a standard iFeature, one HUGE advantage that Sheet Metal punches have, is a planar selection is NOT required. This saves a selection on every single Sheet Metal Punch placement, so we will take care to ensure that we can place all our features from that singular point reference!

**Important Note:** Embossed Sheet Metal features can rarely if ever be flattened on a flat pattern, so one must supply a representation or investigate the Unwrap command.

### Traditional emboss punches

If you're unfamiliar with what an embossed feature is, please look at some traditional examples below. I'll be using this term to mean features that do not completely separate metal from the main piece.

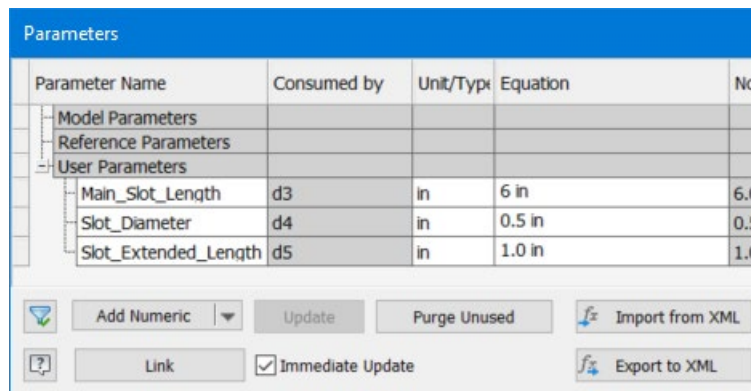


*OUT-OF-THE-BOX EXAMPLES OF EMBOSSED SHEET METAL FEATURES (FRONT VIEW ON THE LEFT, BACKSIDE VIEW ON THE RIGHT)*

## Traditional emboss example

Many of the same techniques that we saw with the compound iFeatures will be directly applicable here. It is critically important that we utilize a singular design sketch as much as possible. In this first emboss example, we'll have a straight-forward time with sketch that will include geometry for a Revolve Join and a Revolve Cut. For a video walkthrough of this example, please see [the video](#) in Appendix A.

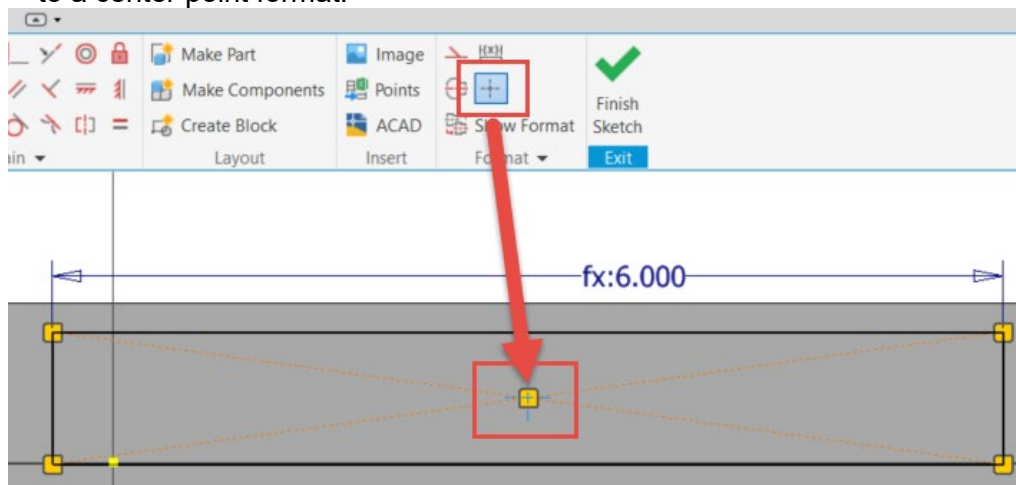
Step 1. After creating and saving the base model geometry, create the necessary parameters that will eventually control the punch.



*CREATE THE PARAMETERS NECESSARY TO CONTROL THE PUNCH*

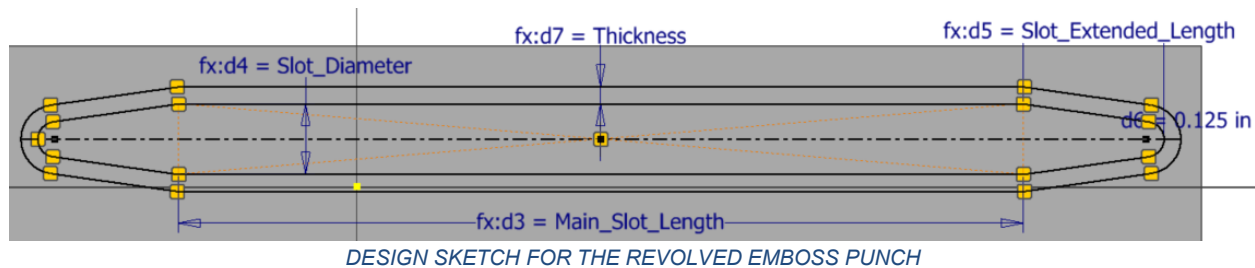
Step 2. Create a sketch that represents the required emboss feature, as well as the cut / hollow that will result of the forming operation. There are several ways that these shapes could be modeled, but there are some general considerations to keep in mind.

- Create a point positioned on the shape such that the feature will fit on the entire part. Be sure that the point is using the center point format, as this will serve as the location point for the published punch. In this example, I drew the main shape with a center point rectangle and I'm using the center point of that shape and converting it to a center point format.

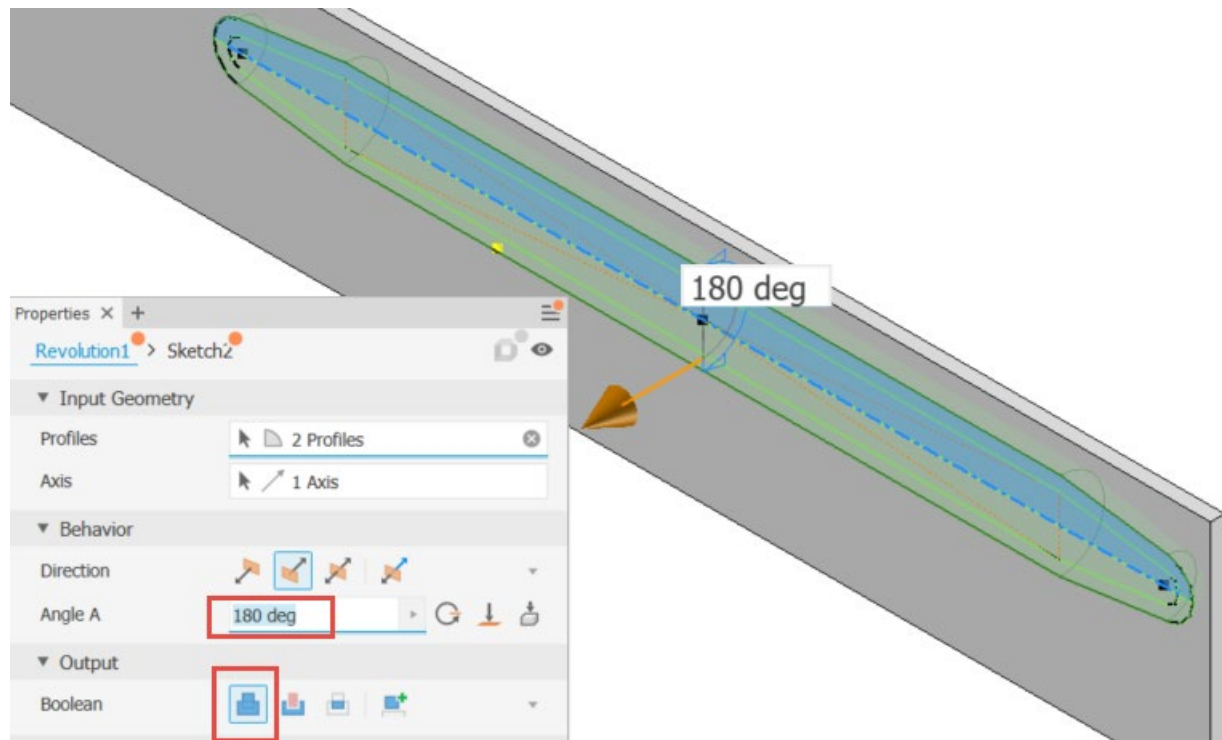


*CONVERT THE CENTER POINT OF THE RECTANGLE TO USE THE CENTER POINT FORMAT*

- Create a centerline through the sketched geometry for use with the linear diameter dimensions and the Revolve feature.
- Ensure that all points contact or can be separated by the sketch centerline, so that all profiles can be revolved properly. Or you can draw in short colinear segments to properly divide the profiles.
- Set any unnecessary geometry to use the construction line format.

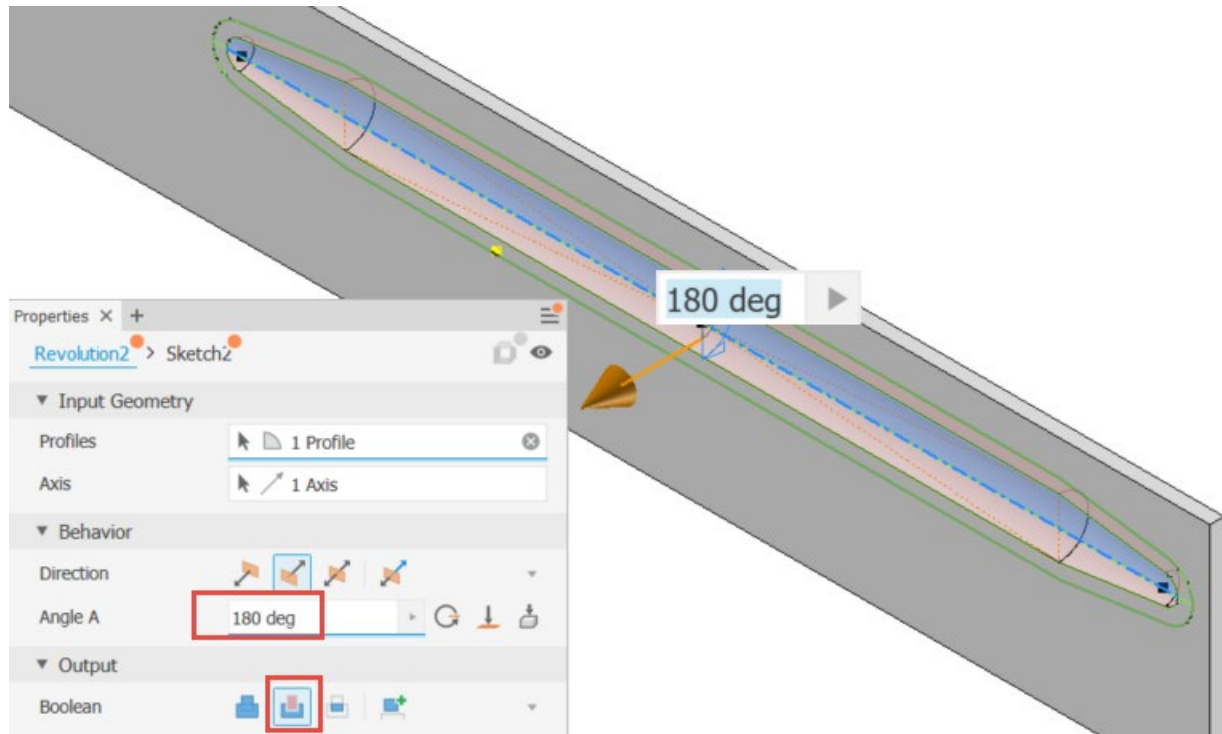


Step 3. Create a Revolve feature that represents the outer shape of the emboss punch. This will be a solid Join, revolved only 180 degrees, as shown below.



SOLID REVOLVE FEATURE AT 180 DEGREES TO GENERATE THE EMBOSS PROFILE

Step 4. Create a cut Revolve feature from the inner profile of the design sketch that will represent the slot geometry. Again, we'll use 180 degrees for the cut.

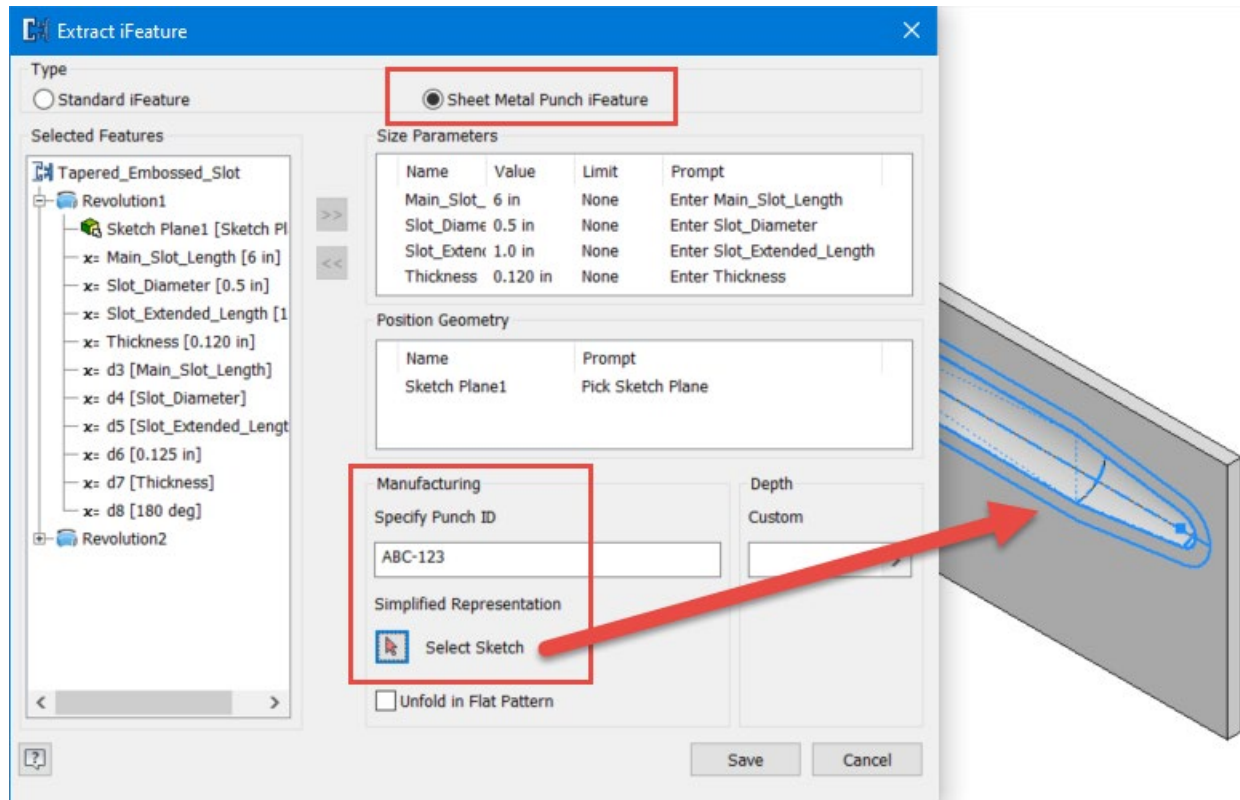


*REVOLVE CUT THE INNER PROFILE TO GENERATE THE SLOT FEATURE*

Step 5. Publish the Sheet Metal Punch in a similar fashion to a standard iFeature. We'll use the same dialog box but will choose the radio button option for the Sheet Metal Punch. Most everything is the same, but there are a couple of specific considerations for Sheet Metal Punch publishing.

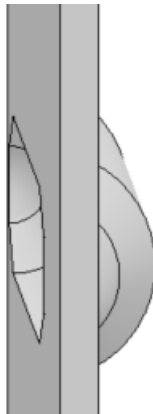
- A center point sketch is required for publishing. If such a sketch is NOT present, the operation will fail, and you'll have to add one.
- Besides naming the feature, you can also set a Punch ID. I STRONGLY recommend this, so punches can be more easily identified with notes and punch tables on drawings.
- If you have a particular view that you'd like for the 2D punch representation, you can set that as well.





EXTRACT THE IFEATURE AS A SHEET METAL PUNCH AND CONFIGURE ANY DESIRED PUNCH INFORMATION

Step 6. Test out the Punch by creating a new sheet metal part and placing center points on a sketch.

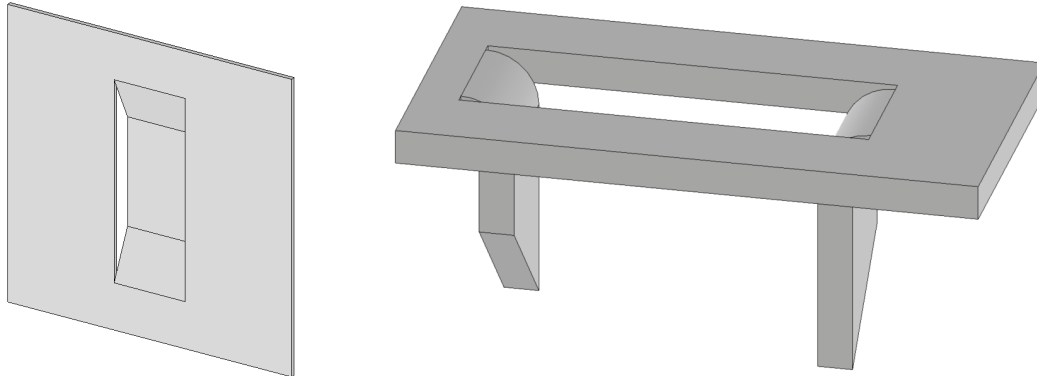


RESULTING SHEET METAL EMBOSS PUNCH IFEATURE



## Open ended formed punches

Open ended formed punches are very similar to traditional embossed punches but have some open ends. This usually means there's a combination of a Cut and Join features to accomplish this goal. While these can sometimes be achieved with a single sketch, most of the time more than one sketch will be required. Please see the following images for some examples of such features.

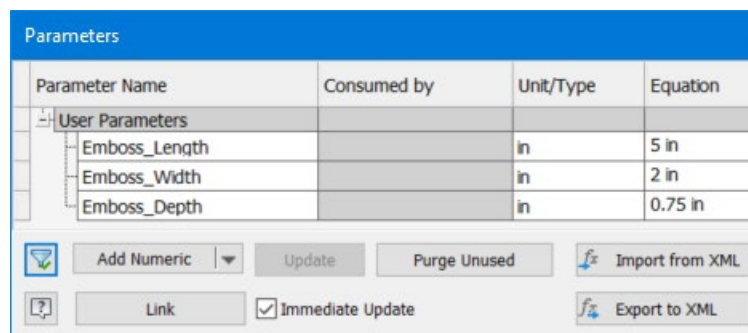


EXAMPLES OF OPEN PUNCHES, A FORMED OPEN-ENDED POCKET ON THE LEFT AND FORMED TABS ON THE RIGHT

## Open ended emboss example #1

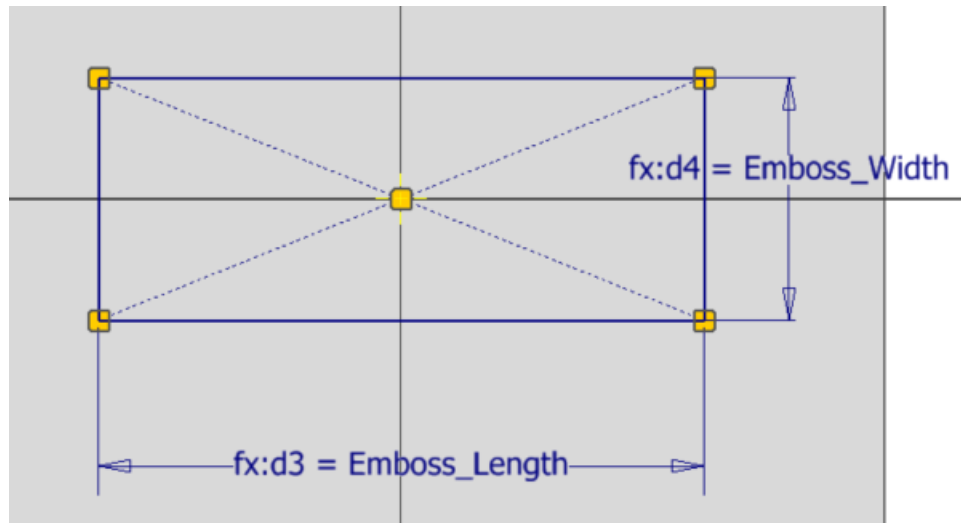
In this first example, we'll take on the open-ended formed pocket, shown on the left in the image just above. This feature is straightforward to model, which is great, but will require second sketch to be easily generated. (If you're interested in trigonometry and fun functions in Inventor, then this could be created with a single sketch :>) For a visual demonstration on building this feature, please see [this video](#) in Appendix A.

Step 1. After creating and saving the base model geometry, create the necessary parameters that will eventually control the punch.



CREATE THE PARAMETERS THAT WILL BE USED BY THE EMBOSS FEATURE

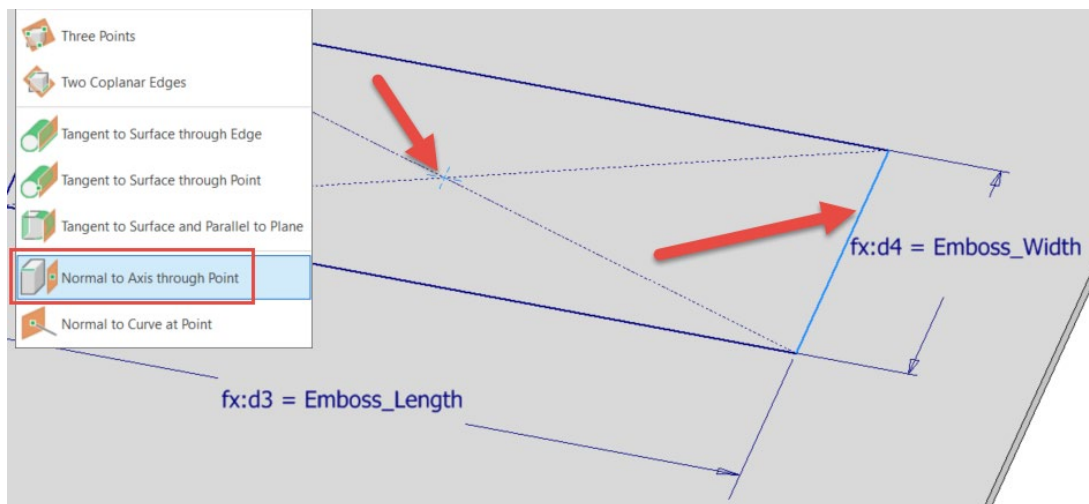
Step 2. Create a sketch of the overall shape of the embossed feature, which will also serve as the cut. In this example, I'm using a center point rectangle sketch feature and am directly utilizing the User Parameters.



*CREATE A SKETCH THAT REPRESENTS THE OVERALL SIZE OF THE EMBOSSED FEATURE AND WILL ALSO SERVE AS THE CUT SKETCH PROFILE*

Step 3. Since this feature would be difficult to create from a single sketch, we'll have to establish a second plane to create the embossed portion of the design. We'll create a work plane based exclusively on the geometry of the sketch. This will allow us to maintain our feature links solely to the initial sketch, thus eliminating unwanted additional placement references.

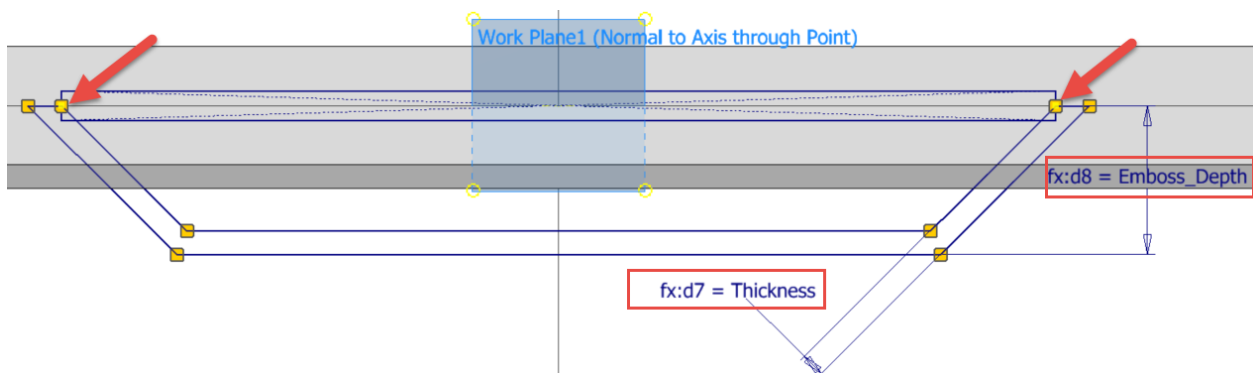
- In this case, we'll utilize the work plane option "normal to axis through Point" with one of the "Emboss\_Width" edges and the center point of the original sketch.



*CREATE A WORK PLANE USING SOLELY GEOMETRY FROM THE FIRST SKETCH*

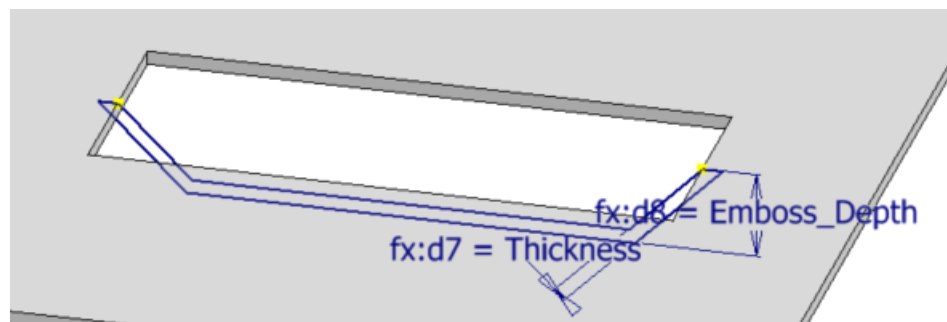
Step 4. Next we'll create a sketch on the new work plane to represent the final emboss shape.

- To define the position of the emboss, project the geometry of each "Emboss\_Width" edge, which should result in a singular dot (NOT a center point formatted point).
- Define the shape and then offset the other side by the "Thickness" parameter.
- Ensure that the profile is closed with the two sets of linework connected with a flat (horizontal) line.
- Set the Emboss Depth as desired, either to the inside or outside position of the pocket.



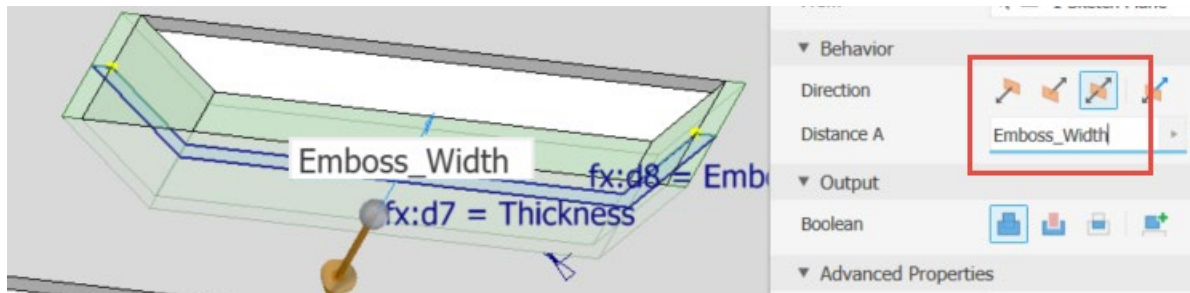
DEFINE THE EMBOSS PROFILE BY ONLY PROJECTING POINTS FROM THE ORIGINAL SKETCH, NOT THE 3D GEOMETRY

Step 5. Utilizing the original sketch, create a Cut feature through the plate.



GENERATE A CUT FEATURE USING THE ORIGINAL SKETCH

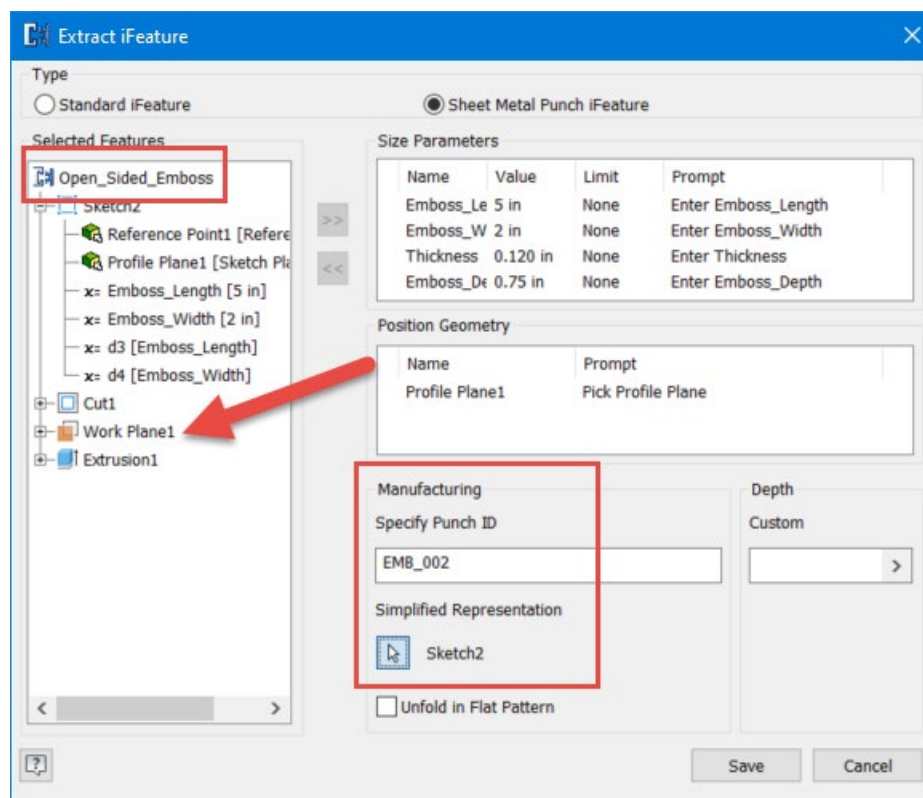
Step 6. Extrude the emboss portion of the feature and be sure to utilize the "Emboss\_Width" parameter, NOT the 3D geometry (for example, do NOT use the "From / to" method AKA between).



EXTRUDE THE EMBOSS FEATURE SYMMETRICALLY USING THE "EMBOSS\_WIDTH" PARAMETER

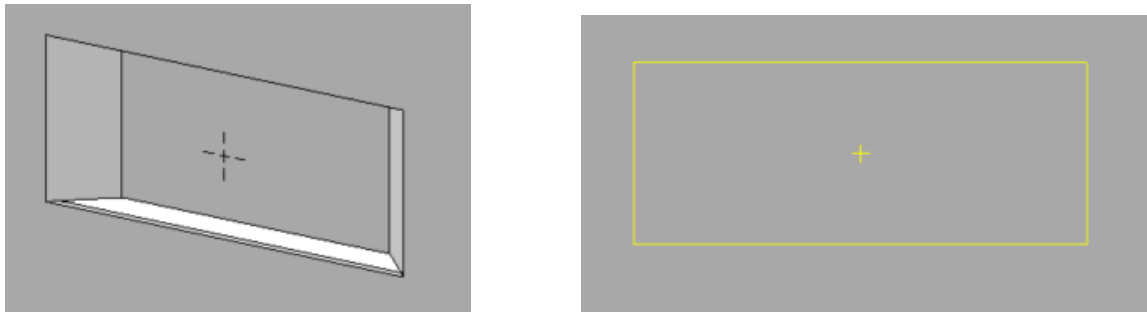
Step 7. Extract the Sheet Metal Punch.

- Be sure to select the Work Plane as one of the features, as this will eliminate one of the required Placement Geometry selections. This is because the work plane is the basis for the second sketch and is generated SOLELY from the original sketch.
- All the previous information is still relevant here, so be sure to name the feature, add a punch ID and set the 2D representation to the initial sketch.
- We'll utilize the simple sketch for the 2D representation for this example. In the next example, I'll demonstrate the technique for adding a more complex 2D representation sketch.



PUBLISH THE SHEET METAL PUNCH USING THE WORK PLANE TO SIMPLIFY PLACEMENT GEOMETRY REQUIREMENTS

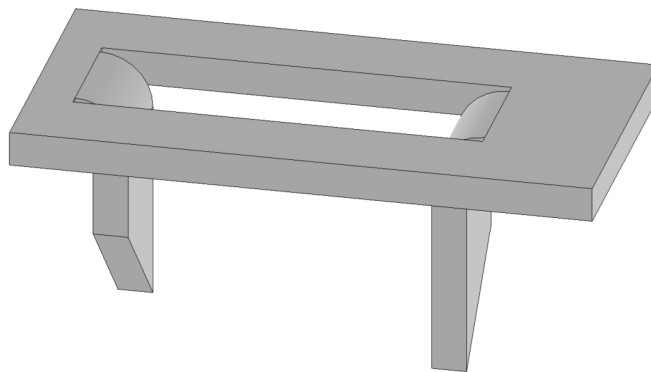
Step 8. Test out the Punch by creating a new Sheet Metal component and face feature. Place some sketch center points and insert the Punch. Note the difference between the formed and flat pattern versions of the Punch.



*RESULTS OF THE OPEN-SIDED EMBOSSED PUNCH, 3D ON THE LEFT, FLAT PATTERN ON THE RIGHT*

## Open ended emboss example #2

In the previous example, we looked at using two sketches to create an embossed feature. That is a great technique to accomplish the goal, but if one is willing to do some math, then a design can be driven from a single design sketch. In this next example, we'll introduce some new concepts that can aid in even more advanced iFeatures and Punches. As a refresher, see the image below for the feature that we're going to be creating next. For a visual demonstration on building this feature, please see [thaialdfasdvideolink](#) in Appendix A.



*A FORMED SET OF "TEETH" GENERATED BY A CUTTING AND FORMING OPERATION*

There is quite a bit that is going on with this feature.

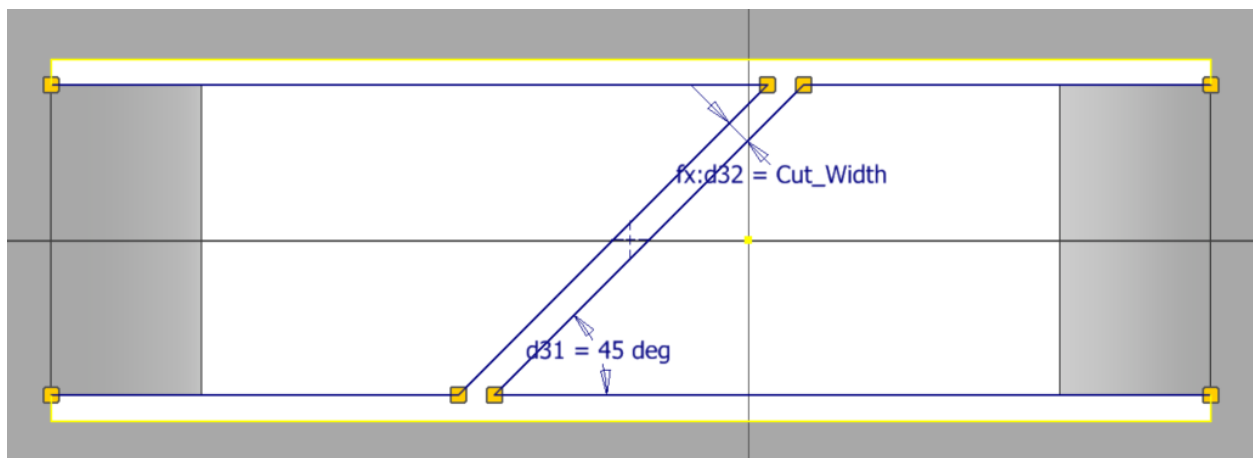
- First of all, this is different than the last feature, as this is more of a forming operation than an emboss.
- If someone wanted to form this feature, vs. just graphically representing the design (as in the last example), there is lots of math that we'll have to compute.
- To properly size the cut, we must take the bend allowances into account.
- The final 2D Punch Representation will be accurate enough to be used to cut out the feature on the sheet.
- We are going to make a design assumption that the teeth chamfers are 45 degrees.

Step 1. After creating and saving the base model geometry, create the necessary parameters that will eventually control the punch.

Parameters			
Parameter Name	Consumed by	Unit/Type	Equation
Model Parameters			
Reference Parameters			
User Parameters			
Cut_Width	d32, Chamfered_Gap...	in	0.03125 in
K_Factor	Bend_Allowance_90D...	ul	0.44 ul
Feature_Bend_Radius	Pattern_Length, Ben...	in	0.0625 in
Tab_Height	Pattern_Length, Tab...	in	0.75 in
Tab_Width	Pattern_Length, d28,...	in	0.375 in
Tab_Length_From_Top	d7	in	Tab_Height + Thickness
Bend_Allowance_90Deg	Pattern_Length	in	( Feature_Bend_Radius + K_Factor * Thickness ) * PI / 2 ul
Chamfered_Gap	Pattern_Length	in	sqrt((Cut_Width) ^ 2 ul * 2 ul)
Pattern_Length	d21	in	(( Tab_Height - Feature_Bend_Radius ) + Bend_Allowance_90Deg ) * 2 ul - Tab_Width + Chamfered_Gap

*CREATE THE PARAMETERS THAT WILL BE REQUIRED TO CONTROL THE PUNCH*

There is a bunch of math in the Equation fields for these user parameters and it may not appear to be obvious how things are being computed. To help better understand the situation, please see the image below which should help.



*IMAGE OF THE DESIRED FLAT PATTERN RESULT*

As you can see in the image, there would be some overlap if the teeth were straight cut. Since they are cut at 45 degrees and I wish to utilize the same "Cut\_Width" all around the design, we must do some math. We must subtract the "Tab\_Width", since the tooth

angle is 45 degrees we have to subtract  $\frac{1}{2}$  "Tab\_Width" from each tooth and must perform some  $A^2 + B^2 = C^2$  trigonometry to compute the straight-line distance for that diagonal sheet cut. That is covered in the Equation Field for the parameter "Chamfered\_Gap", as A & B legs would be the same, since this is a 45-degree angle.

Additionally, to calculate flat size, we'll have figure out the bend allowance for the final formed shape. Most of the time, we don't have to worry about that as Inventor handles that when building the flat pattern, based on whatever unfold criteria being specified. However, because the 2D representation must be accurate enough to cut from, we'll have to calculate that allowance ourselves. To simplify things for this discussion, we'll utilize the K-Factor and to familiarize ourselves with that concept, please see the image below.

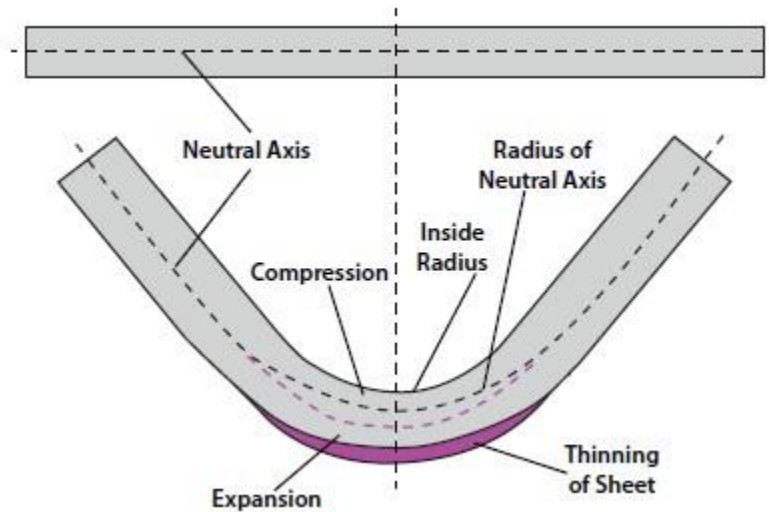


IMAGE SHOWING A COMMON FORMING OPERATION. K-FACTOR IS THE RATIO OF THE NEUTRAL AXIS BETWEEN THE INSIDE AND OUTSIDE RADII

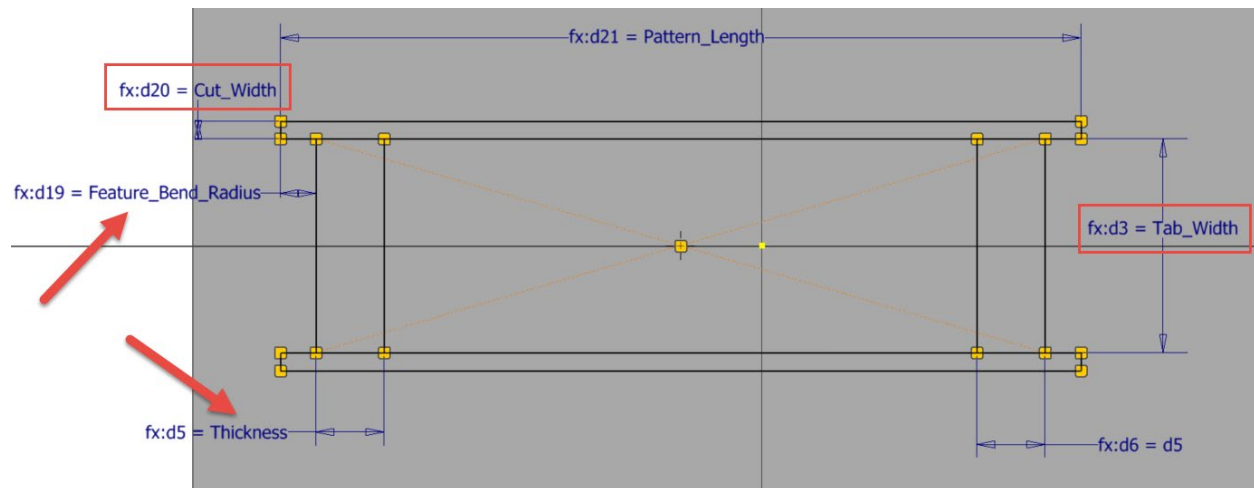
The K-Factor is the ratio location of the Neutral Axis between the inside and outside bend radii. For example, a K-Factor of 0.44 would mean the Neutral Axis is 44% of the way from the inside radius traveling toward the outside radius, or 44% of the thickness. If we add the inside bend radius to the K-Factor multiplied by the thickness, we can achieve the radius for the Neutral Axis. Multiplying this with the bend angle will give us the bend allowance. Since the bend angle is 90 degrees, we can multiply the Neutral Axis radius by  $\pi / 2$ . This is computed with the "Bend\_Allowance\_90Deg" parameter.

We'll revisit more of this math after Step 2, when we start to set up for the feature.



Step 2. Create a sketch that will act as the placement sketch and rough out the features. There are several interesting components to this sketch.

- This is top-down view of the feature. We must model the feature this way, as the Bend feature is not available for use when publishing Punches.
- The “Tab\_Width” and “Cut\_Width” parameters work together to control the overall flat pattern width of the feature.
- Since the “teeth” will be modeled downward from this vantage point, we need to move the teeth away from the ends of the pattern by the distance of the “Feature\_Bend\_Radius”, which is the inner radius of the bend.
  - o This is why the “Pattern\_Length” parameter contains the mathematical phrase  $(\text{Tab\_Height} - \text{Feature\_Bend\_Radius} + \text{Bend\_Allowance\_90Deg})$  as we must remove the inner bend radius distance before the accurate bend allowance distance can be added.
- To create the “teeth” features, we’ll have to utilize the Extrude command, so we’ll simply denote the tooth location and size with a rectangle using the “Thickness” parameter.

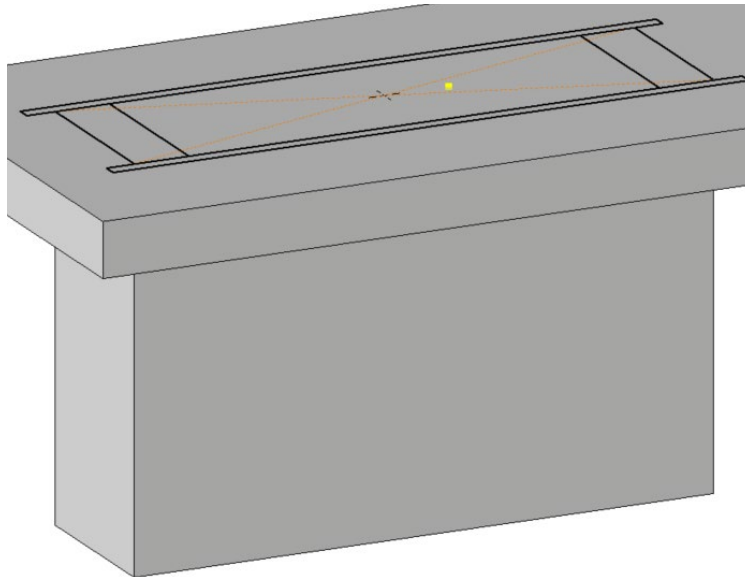


DESIGN SKETCH FOR PLACEMENT AND A TOP-DOWN VIEW OF THE FEATURE. THE PARAMETERS WORK IN CONCERT TO CALCULATE THE ACCURATE “PATTERN\_LENGTH” PARAMETER FOR THIS FEATURE

Step 3. Extrude the solid portion of the tooth geometry. Since we’re going to cut away material in a little while, this should only be a rough feature, as you can see in the following image. Use the “Tab\_Length\_From\_Top” parameter for the extrusion length.

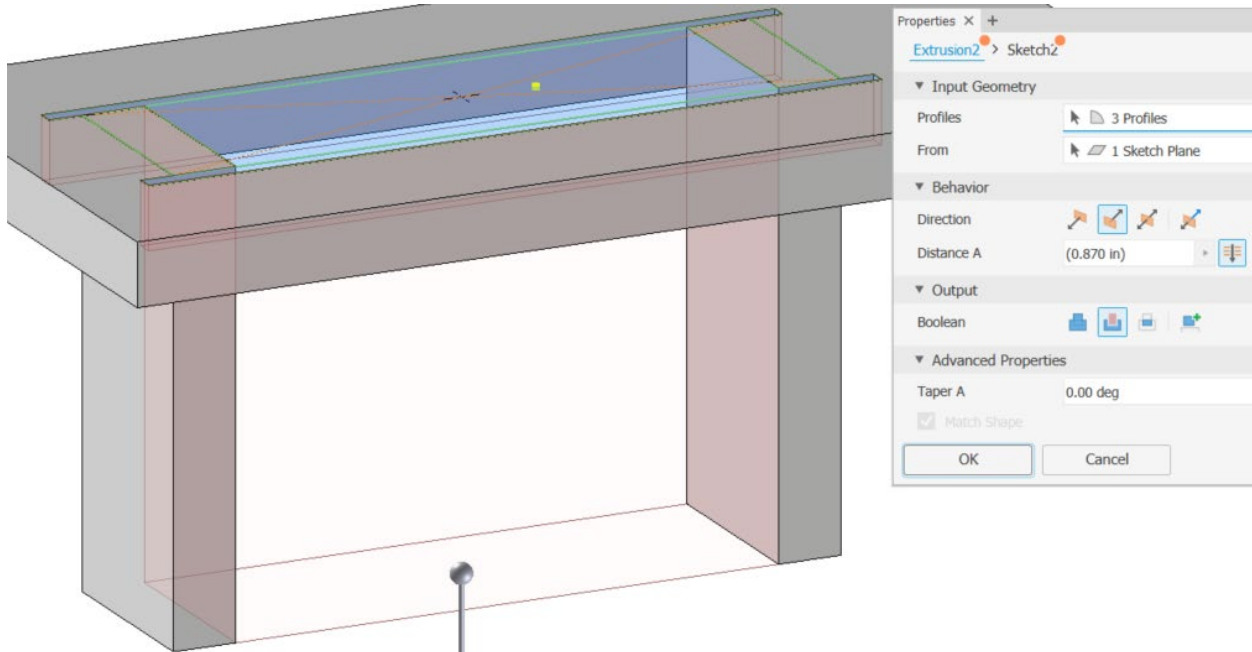
**Bonus Tip:** If you click on the apply button (the plus sign at the lower left corner of the Extrude dialog box) you can execute the solid extrude but leave the dialog box open to execute the cut extrude in Step 4. This will also “Share” the sketch for you. For a visual demonstration of this technique, please see [the video](#) in Appendix A.





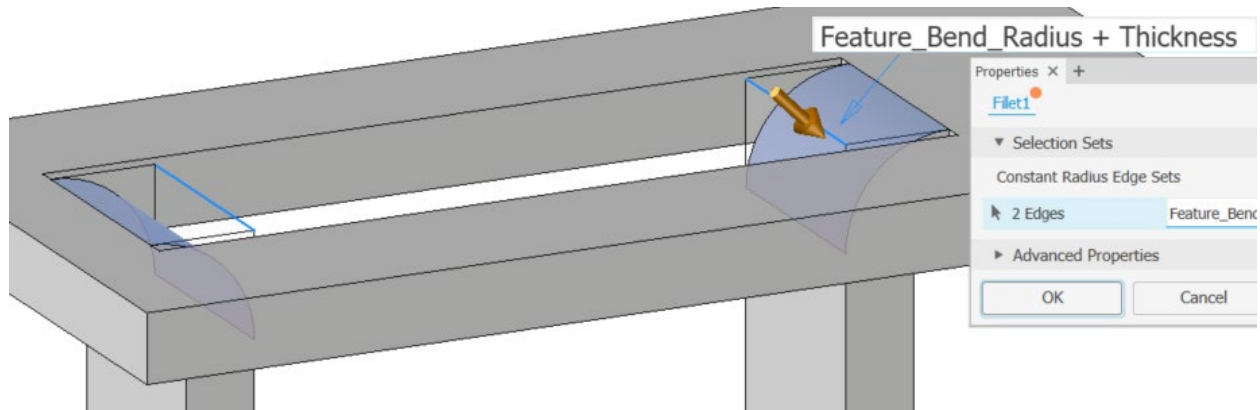
*SOLID EXTRUSION OF THE TOOTH PROFILE. SINCE MATERIAL WILL HAVE TO BE CUT AWAY, THIS INITIAL SOLID CAN BE QUITE ROUGH*

Step 4. Using the same sketch, create a Cut extrude feature to eliminate the unnecessary material.



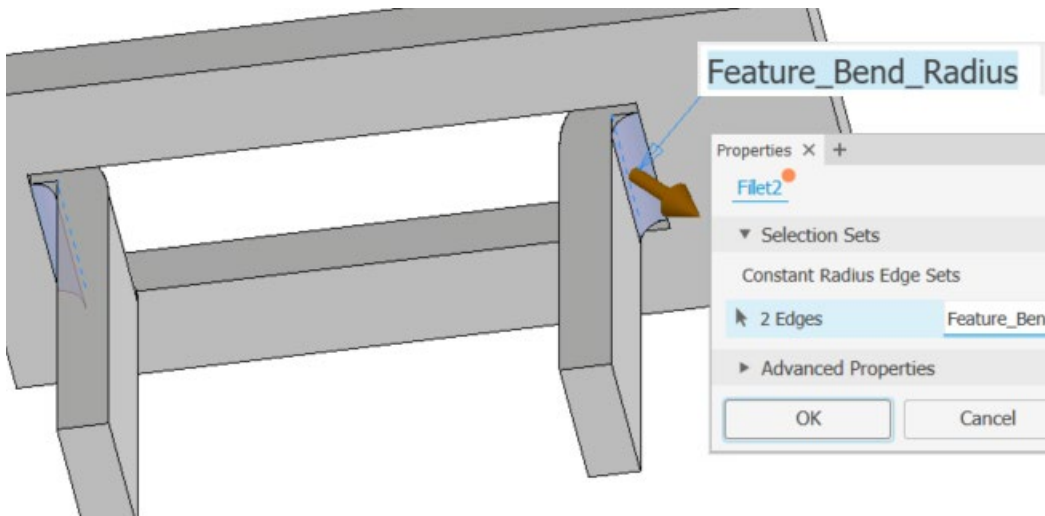
*GENERATE A CUT EXTRUDE FEATURE TO CLEAR OUT ALL THE UNNECESSARY MATERIAL*

Step 5. Apply a pair of fillets on the top surface edge of the teeth. Utilize the equation “Feature\_Bend\_Radius” + “Thickness” to define the radius.



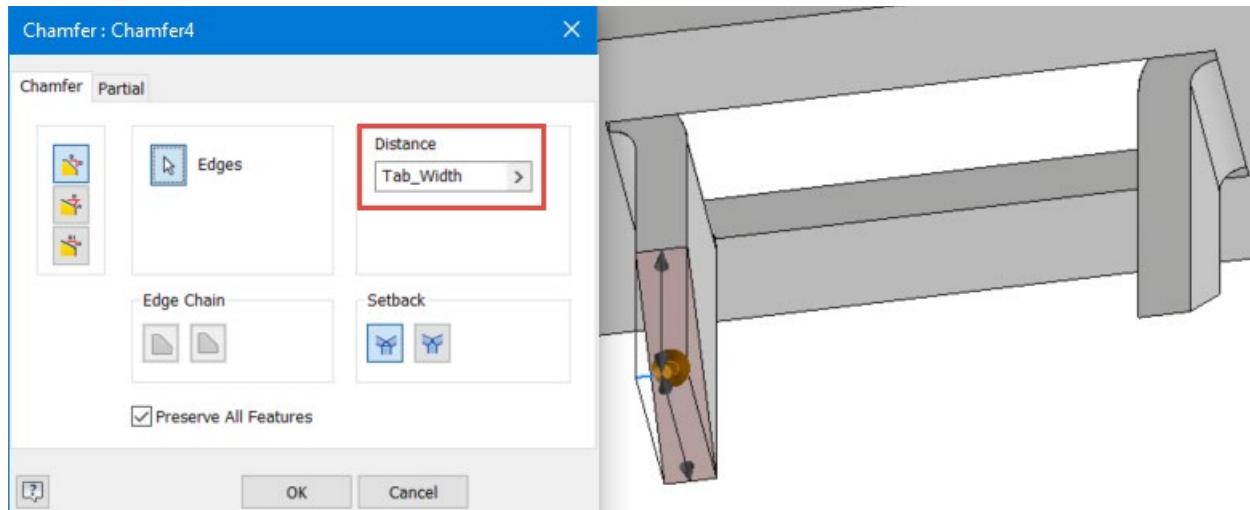
*APPLY TOP SIDE FILLETS USING FEATURE\_BEND\_RADIUS + THICKNESS PARAMETERS*

Step 6. Apply a pair of fillets on the bottom edge surface of the teeth. The radius for these features is simply the “Feature\_Bend\_Radius”.



*APPLY BOTTOM-SIDE FILLETS USING THE FEATURE\_BEND\_RADIUS PARAMETER FOR THE RADIUS*

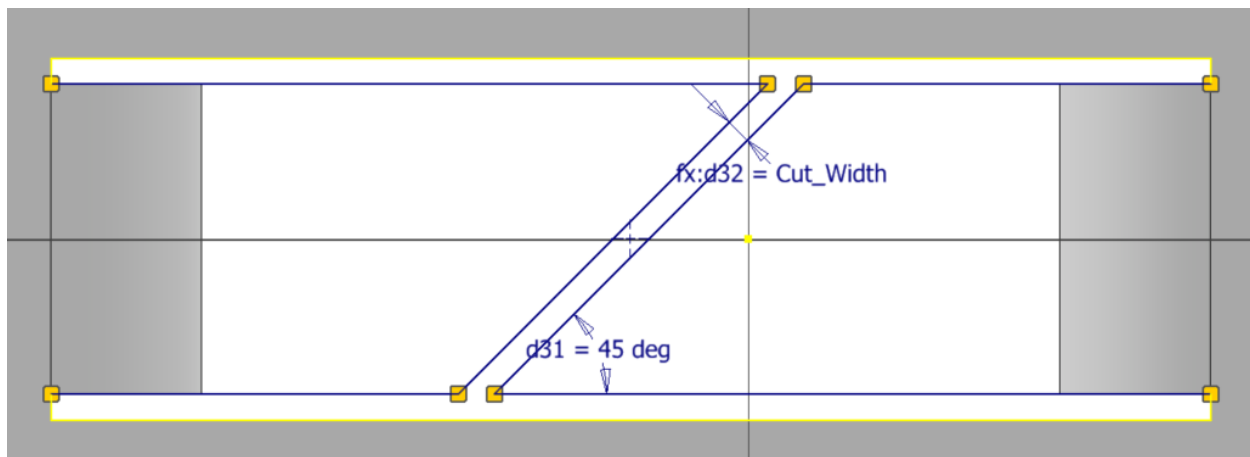
Step 7. Apply opposing fillet features at the bottom ends of the tabs using the “Tab\_Width” parameter value for the size. Be sure the features are mirrors of each other.



*CREATE OPPOSING CHAMFERS TO FINALIZE THE TOOTH GEOMETRY*

Step 8. Now that the feature is modeled, we need to prepare the 2D Representation. Unfortunately, the initial sketch won't work for creating this view, as it will not allow for the accurate cutting of the tooth geometry. Therefore, we'll have to model a second sketch.

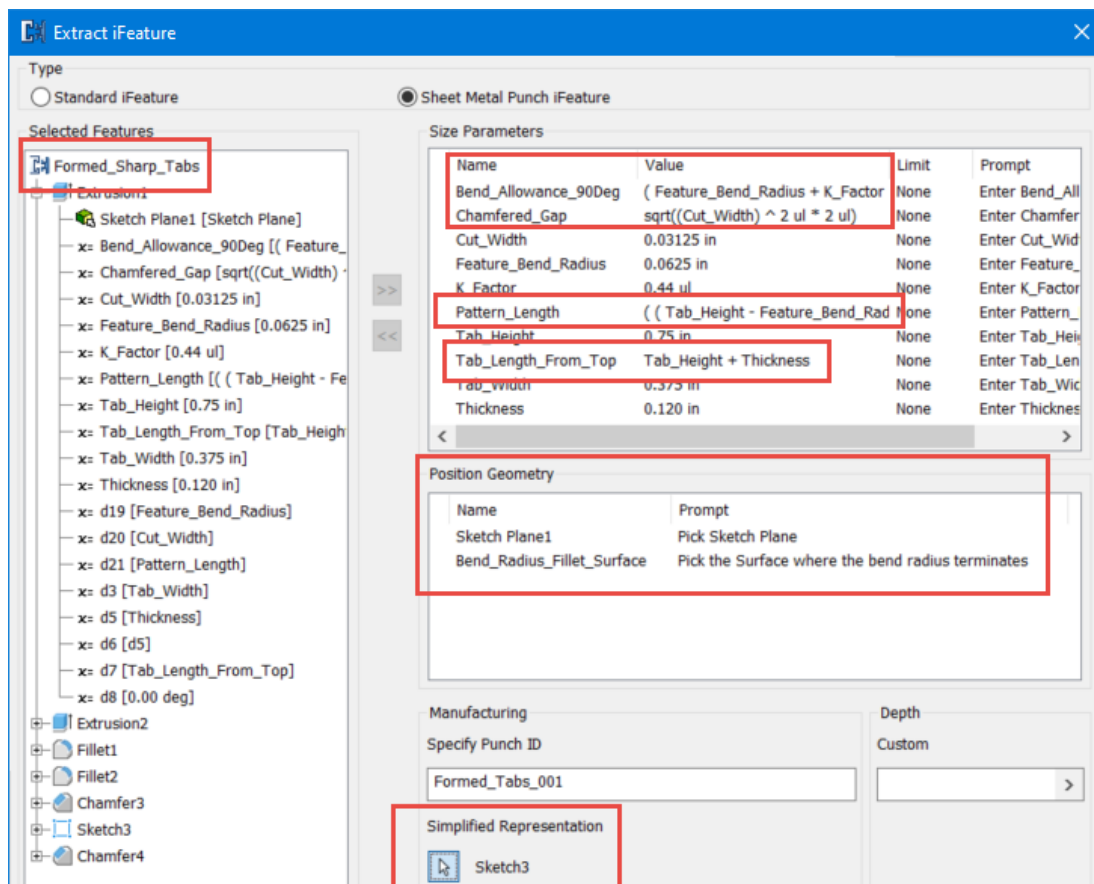
- Make the initial sketch visible and project the other cut edges from that sketch.
- Create new sketch geometry that represents the tooth geometry and is separated by the "Cut\_Width" parameter.
- If you're uncertain about the sketch, measure values and compare those measurements against the values in the parameter table.
- (Optional) rename the sketch into something more useful, to identify it more easily.



*CREATE A NEW SKETCH THAT WILL REPRESENT THE ACTUAL TOOTH GEOMETRY AND BE USED FOR THE 2D PUNCH REPRESENTATION*

Step 9. Publish the Sheet Metal Punch. There are several items to consider with this Punch.

- Rename the iFeature, select features and apply a Punch ID as normal.
- There are several parameters that have been added to the list that we'll have to remove. These parameters contain all the equations necessary to properly build the feature, so we wouldn't want end-users to accidentally manipulate those fields. Even more importantly, these equation parameters will cause the placed iFeature to fail, so they CANNOT be added. The parameters to be removed are:
  - o Bend\_Allowance\_90Deg
  - o Chamfered\_Gap
  - o Pattern\_Length
  - o Tab\_Length\_From\_Top
- There are two entries for the Position Geometry and they both must be used. The "Face" is required because the fillet feature needs to know which host component surface to use when generating that fillet.
  - o Be sure to name the "Face" something helpful. I'll use "Bend\_Radius\_Fillet\_Surface"
  - o You can also add in a more useful prompt as well.
- Select the sketch from Step 8 for the "Simplified Representation".

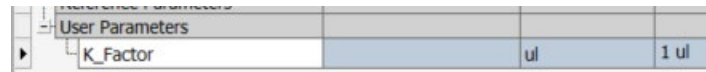


REMOVE EQUATION BASED PARAMETERS, RENAME FIELDS AND SELECT A SIMPLIFIED SKETCH WHEN PUBLISHING

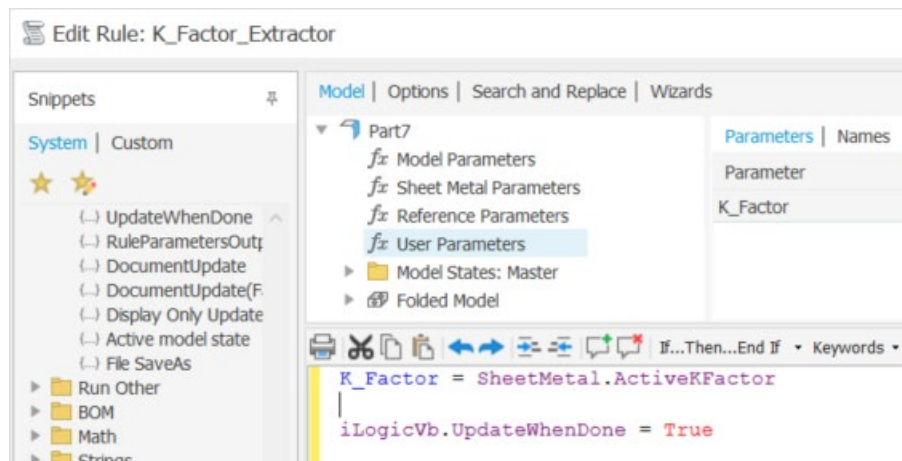
Step 10. Test out the iFeature on a new Sheet Metal component. When testing out the previous iFeatures and Punches, the steps were relatively straight forward. However, due to the desired accuracy of this particular Punch feature, namely the desire for an accurate flat pattern representation for cutting, there are some important considerations required for placing this feature.

- Create the model geometry and the center point sketches as you normally do.
- This feature requires a K-Factor to compute the sizing correctly. To accomplish this, we must do the following.
  - o Using iLogic, extract the current Sheet Metal Rules' K-Factor and assign it to a User parameter.
  - o Link the Punch's K-Factor value to the User Parameter K-Factor value.
- To flatten this feature, fillets were used on both the top and bottom-side of the features. The fillet on the bend radius side needs to reference a surface to generate the fillet, so an additional placement selection will be required.

In this scenario, the iLogic rule will be quite straightforward. First we'll create the User Parameter "K\_Factor", set it to use "Unitless" units and then use the rule to extract the current value. For a copy of the iLogic rule, please see Appendix B.

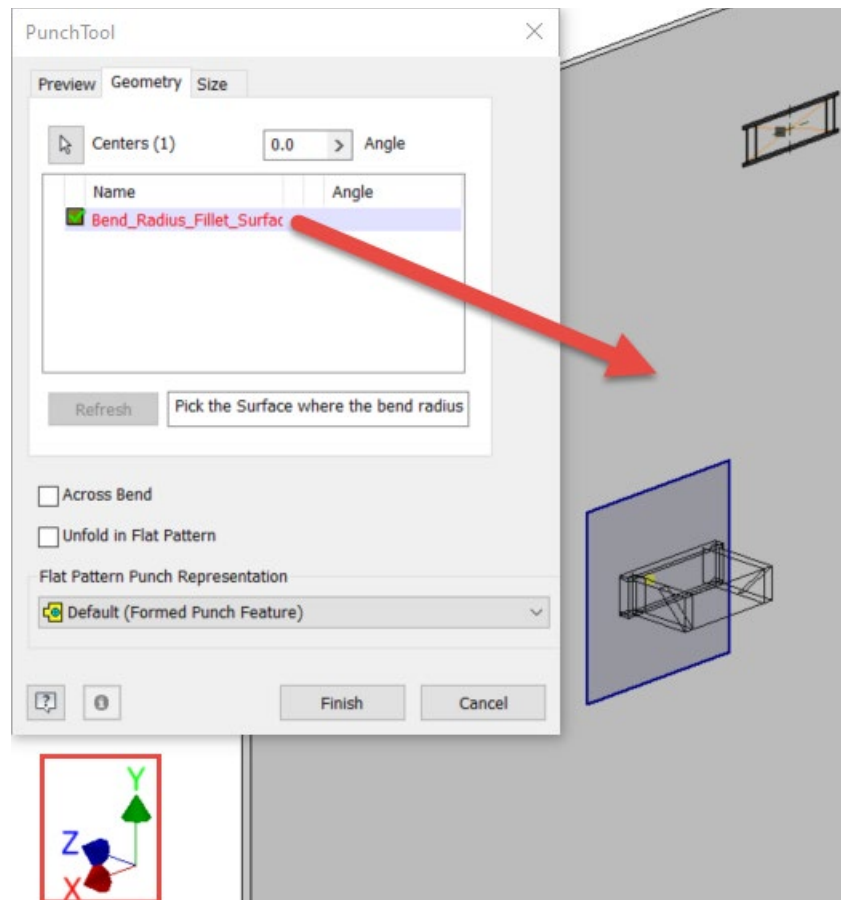


CREATE A "K\_FACTOR" USER PARAMETER IN THE HOST COMPONENT



CREATE AN ILOGIC RULE THAT WILL EXTRACT THE CURRENT K-FACTOR AND ASSIGN THAT VALUE TO THE K\_FACTOR PARAMETER

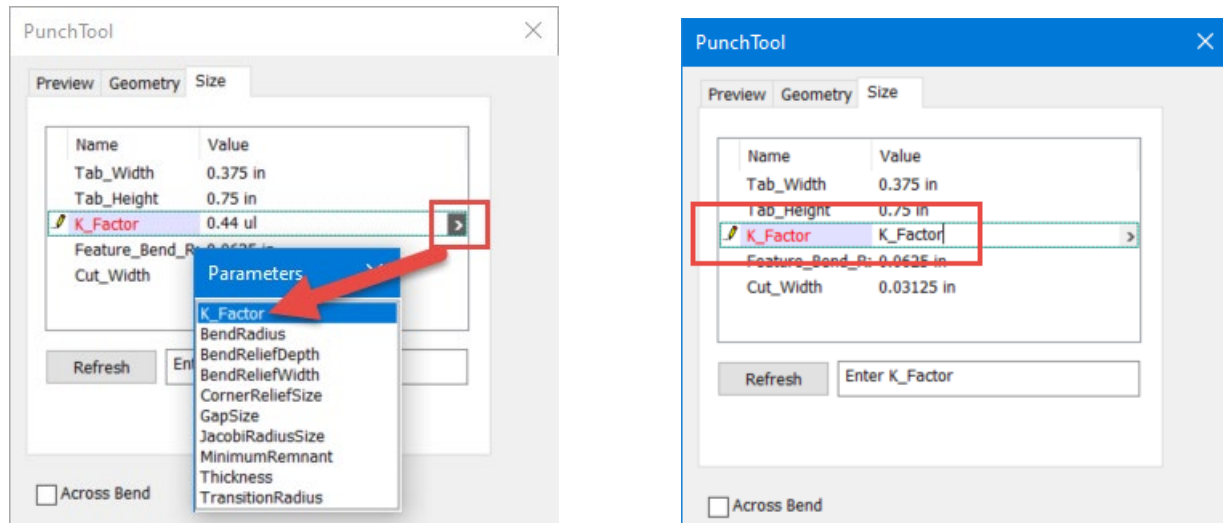
Place the Punch on the part as you normally do and choose the surface on the opposite side of the placement sketch for the fillet surface selection.



*SELECT THE “BACK” SIDE (OPPOSITE THE PLACEMENT SKETCH) SURFACE FOR THE FILLET SURFACE SELECTION*

When we published the Punch, the internal “K\_Factor” value was 0.44. However, in any given host component, the K-Factor could be a different value. Not only that, if an end-user decides to make changes to the material or the unfold criteria, the K-Factor could change in-session. Therefore, we’re going to tie the internal Punch “K\_Factor” to the host component’s “K\_Factor”.

- To accomplish this, switch to the Size tab of the Punch tool and click on the arrow icon at the right-hand end of the “K\_Factor” parameter’s value field.
- Choose the “List Parameters” option just like with normal Inventor modeling.
- Select the “K\_Factor” parameter from the list and now the values are tied together.
- We’ll see this again in the next section when we link pattern iFeatures to the host components, which will likely be more common.



*LINK THE INTERNAL PUNCH PARAMETER TO THE CORRESPONDING PARAMETER IN THE HOST COMPONENT*

Voila! The Punch has now been inserted and is linked. We can now change the material and / or unfold criteria and the Punch feature will update. Take some time to test out this functionality to become more comfortable with it.

**IMPORTANT NOTE:** When switching the unfold criteria for a Sheet Metal component, the rule may not automatically run, so you may want to set a trigger to run the “K\_Factor\_Extractor” rule every time the component is saved or manually run the rule after an unfold criteria change.

## A word on utilizing fillets with emboss punches

I wanted to offer a quick word on using fillets with embossed or formed iFeatures or Sheet Metal Punches. While these can make the features look realistic, they are often unnecessary and often require additional placement references. For example, if we look at the open-sided emboss feature (a refresher image is below), the Inventor Sheet Metal Flat Pattern tool is incapable of flattening that shape, as it is a deformation vs. a fold. Therefore, I typically leave fillets off most iFeatures and Punches that I author.

In rarer situations where the fillets assist with an operation, like in the example that we just completed, then the fillets are absolutely required, and one should make the placement geometry selections as straightforward for the end-user as possible.

As with most things with solid modeling, put a little thought into how much accuracy is required for the desired design and make selections accordingly. One can introduce more complexity into the design later, so I advise starting with a simple a model as possible.



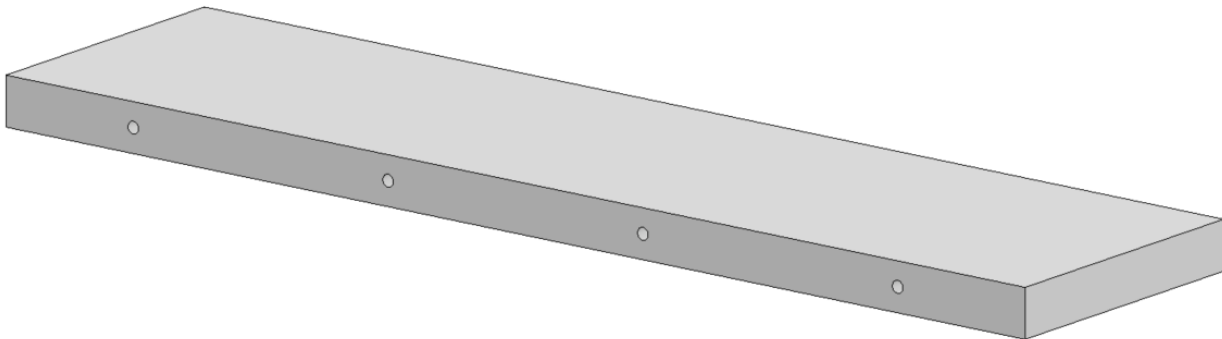
## Internal Patterns in iFeatures

The idea of utilizing a pattern within an iFeature may not be an option that many users consider, but it can be a great way to complete redundant patterns, such as the holes on a fluted pipe. Or another example would be a woodworking company who gets a generic model of a cabinet, perhaps a STEP file, and then must add the necessary manufacturing detail for the programming group. The overall idea of creating the pattern is the same, but there are a couple of important items to consider.

- These patterns are internal the iFeature and therefore NOT recognized as feature patterns for an assembly. This means that components will not be able to use the iFeature pattern as an associative pattern. The parameters from the part would have to be linked into the assembly to drive any component patterns.
- It is best to keep all references internalized as possible for placement selection later. In the following example, we'll even use an internal sketch for the pattern direction versus utilizing a reference 3D surface or edge.
- In this example, I will be placing 2 holes evenly spaced from each end and then equally space the remaining quantity of holes between these two end holes. There are, of course, other types of patterns that could be created (i.e., a simple spacing pattern, not holes on each end).

## Setting up the patterned iFeature

Many of the techniques that have already been explored are very applicable to the creation of a pattern iFeature. We'll step through the creation of the following iFeature pattern consisting of holes along the edge of a board.



*EXAMPLE PART WITH THE PATTERN IFEATURE APPLIED*

Step 1. After creating and saving the base model geometry, create the necessary parameters that will eventually control the pattern of features.

- In this example, we list the desired quantity of holes and a formula in the "Hole\_Spacing" parameter will generate the necessary spacing between holes.
- I'm not specifying the hole diameter or depth, for simplicity's sake, but those parameters could be added and utilized, if desired.

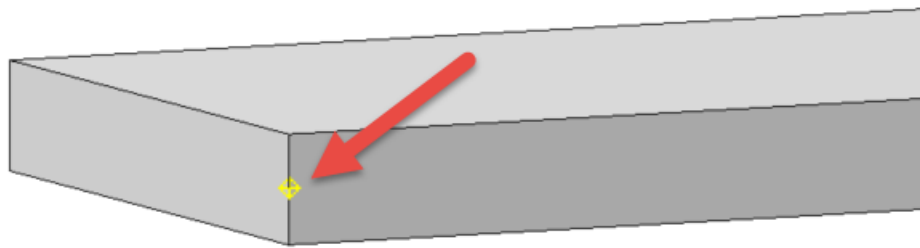


Parameters			
Parameter Name	Consumed by	Unit/Type	Equation
Model Parameters			
User Parameters			
OA_Length	d0, Hole_Spacing	in	18 in
End_Dist	d4, Hole_Spacing	in	3 in
Hole_Spacing	d17	in	$(OA\_Length - End\_Dist * 2) / (Hole\_Qty - 1)$
Hole_Qty	d15, Hole_Spacing	in	4 in

☒ Immediate Update

PARAMETERS REQUIRED TO BUILD AND CONTROL THE PATTERN

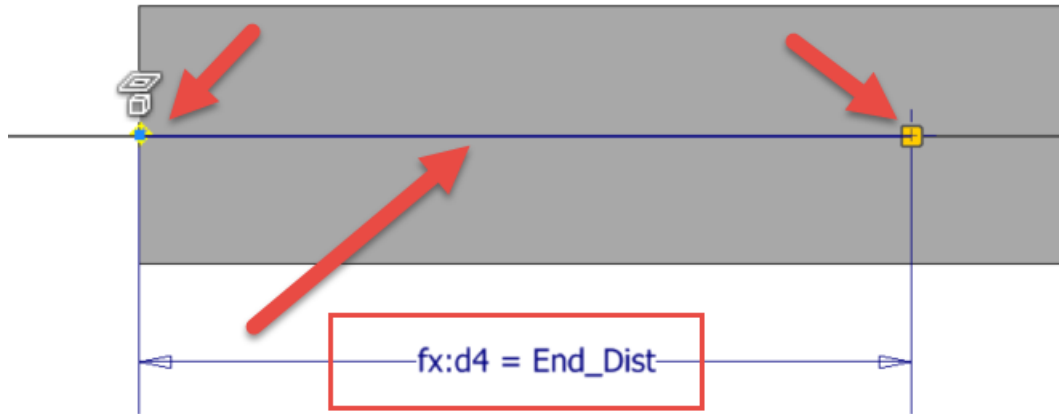
Step 2. Choose a reference point on the geometry to use as the location for the end of pattern. In this example, I'll choose the midpoint of the left thickness edge. This will serve as the measurement point for the end distance to the first hole, as well as the locator point when placing the iFeature.



CREATE A WORK POINT AT THE MIDPOINT OF THE LEFT VERTICAL EDGE OF THE THICKNESS FACE

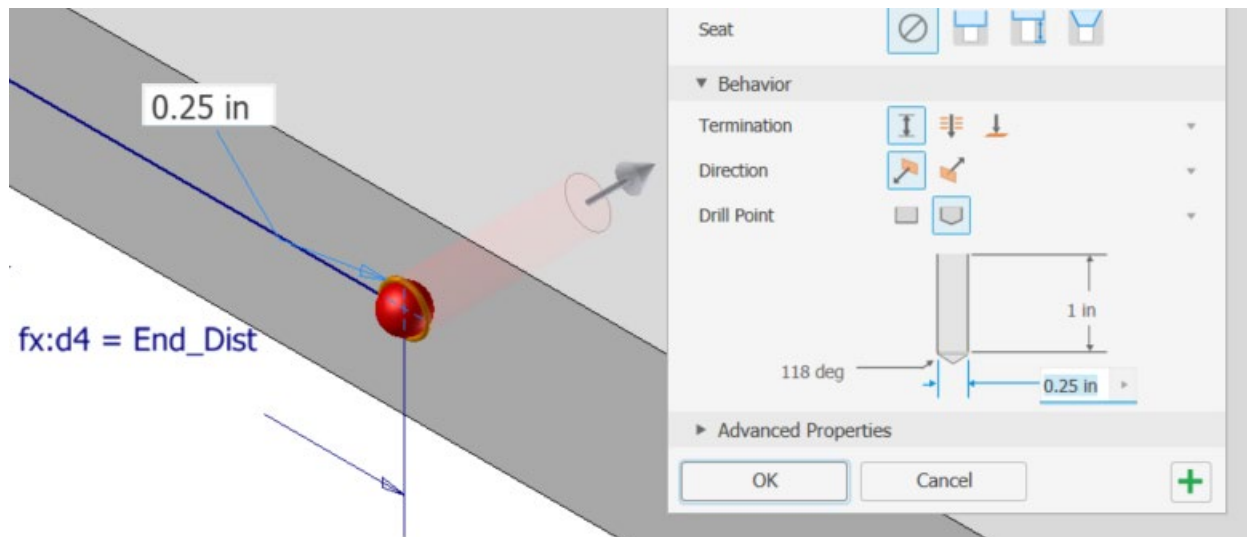
Step 3. Generate a sketch on the long thickness surface that is touching the newly minted work point.

- Project the work point onto the sketch and convert it to a standard point (not center point format). This will allow us to utilize the point in the sketch but will not automatically generate a hole center when using the hole command later.
- Sketch a horizontal line directly to the right, starting from the projected point. Dimension the distance using the "Edge\_Dist" parameter and this will serve as the location point for the first hole.
- Either select the rightmost endpoint of the newly drawn line or create a new point at the right-hand end. Be sure that this is a center point formatted point.



*IFEATURE SKETCH CONTROLLING THE ORIENTATION AND POSITION OF THE FIRST HOLE*

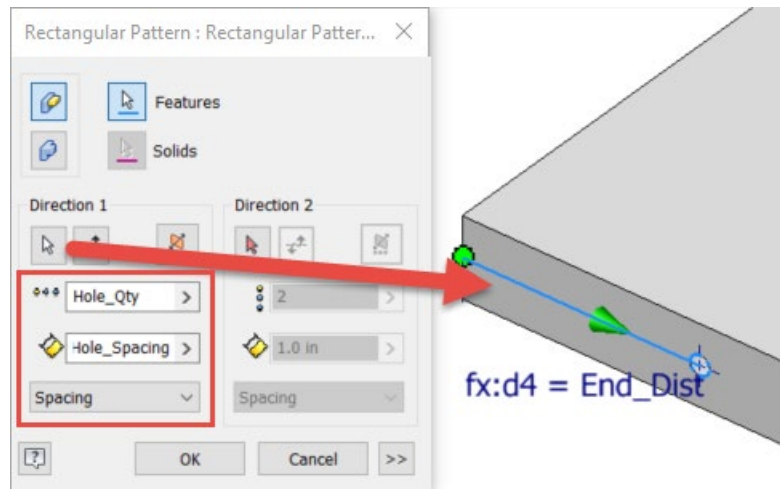
Step 4. Finish the sketch and create a new hole feature. In this example, I'm using a generic hole diameter of 0.25 in with a depth of 1 in.



*GENERATE THE HOLE FEATURE THAT WILL BE USED FOR THE PATTERN*

Step 5. Share the sketch and generate the feature pattern.

- Select the hole feature as the basis for the pattern.
- Use the sketch line for the direction of the pattern.
- Set the feature quantity to the "Hole\_Qty" parameter.
- Set the feature spacing value to the "Spacing" parameter.



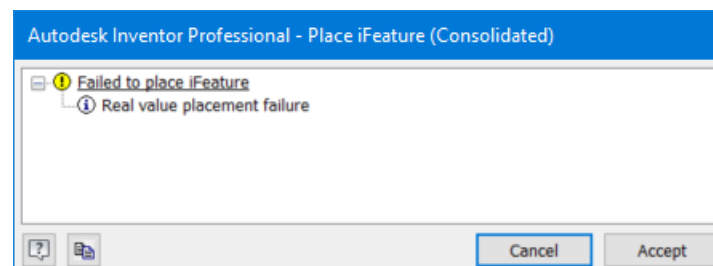
CREATE A RECTANGULAR PATTERN OF HOLES USING THE SKETCH LINE AND “HOLE\_QTY” AND “HOLE\_SPACING” PARAMETERS

Step 6. Publish the iFeature. We can control the parameters and rename the features as in every other iFeature publishing operation, but there are some specific items that I want to call out.

- As I mentioned in the last example, we must remove any parameter that contains an equation vs. a value, in this case the “Hole\_Spacing”. The iFeature can still use this parameter for controlling the pattern, but we won’t expose it for editing. Besides the obvious risk of allowing a user to violate the design intent of the iFeature, Inventor won’t allow an iFeature with a formulaic editable parameter to even be placed. If you see an error message, like the one below, return to the base model and republish without exposing those parameters.

Name	Value	Limit	Prompt
End_Dist	3 in	None	Enter End_Dist
Hole_Qty	4 ul	None	Enter Hole_Qty
Hole_Spacing	$(OA\_Length - End\_Dist) \div Hole\_Qty$	None	Enter Hole_Spacing
OA_Length	18 in	None	Enter OA_Length

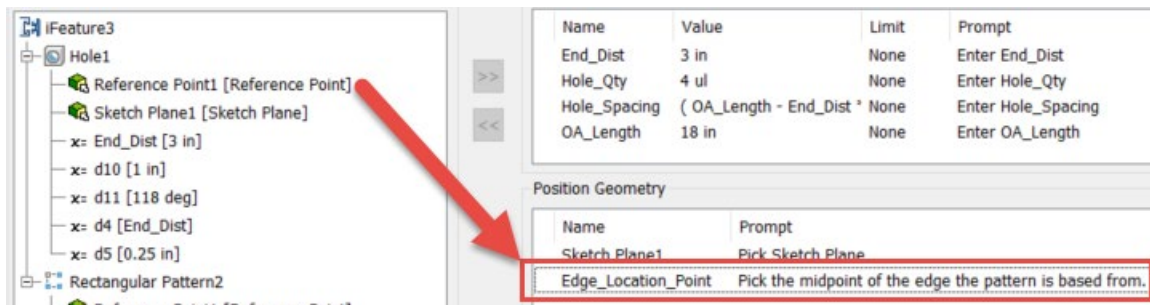
“HOLE\_SPACING” PARAMETER SHOULD BE REMOVED PRIOR TO PUBLISHING THE IFEATURE



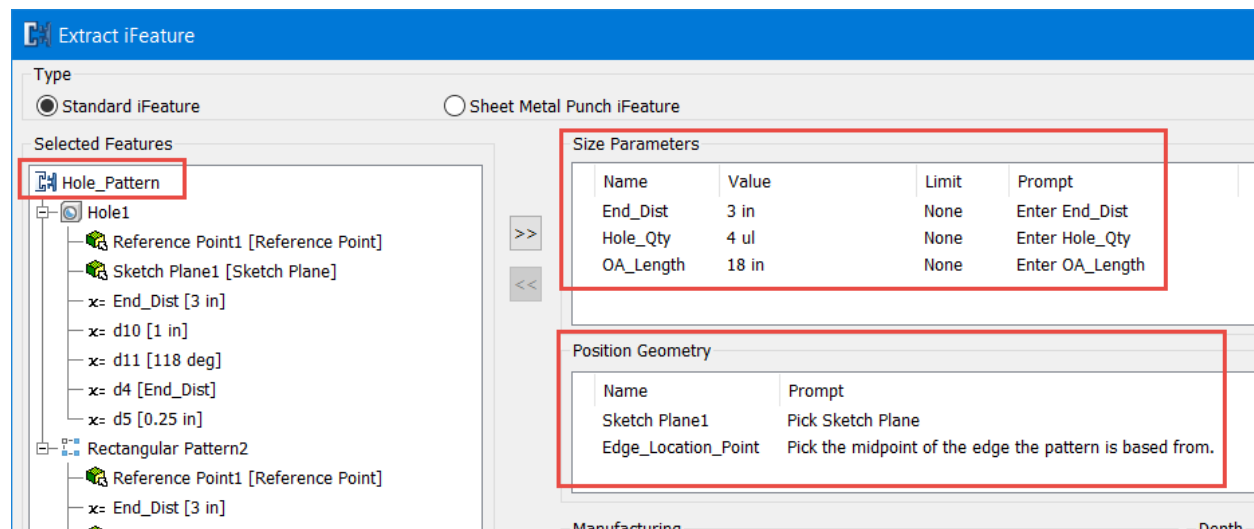
ERROR MESSAGE WHEN PLACING AN IFEATURE CONTAINING EQUATION-DRIVEN PARAMETERS

- Be sure to add the “Reference Point1” into the Position Geometry area. By default, only the sketch plane will be present, but by placing the reference point, the placed iFeature can be located with two selections and not require a sketch-edit placement. To add this to the Position Geometry, simply double left-click on “Reference Point1”. Feel free to rename these as desired.

**Bonus Tip:** One can left click, hold and drag the Position Geometry to rearrange the order of the selections for the end user. For example, the “Sketch Plane1” can be moved above the “Reference Point1”, so that the end user will pick the plane first.



*DOUBLE LEFT-CLICK THE “REFERENCE POINT1” TO ADD TO THE ROSTER OF POSITION GEOMETRY AND RENAME AND REPOSITION AS DESIRED*



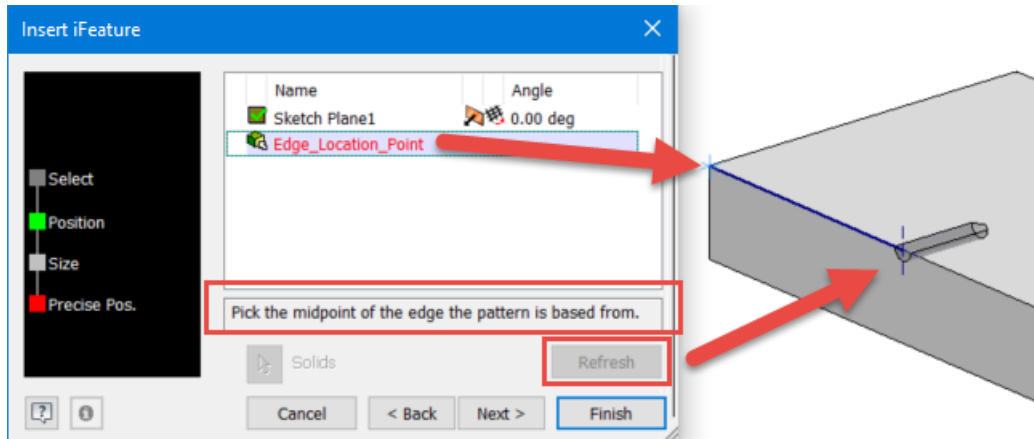
*PUBLISH THE IFEATURE WITH THE SETTINGS AS SHOWN*

## Placing a patterned iFeature

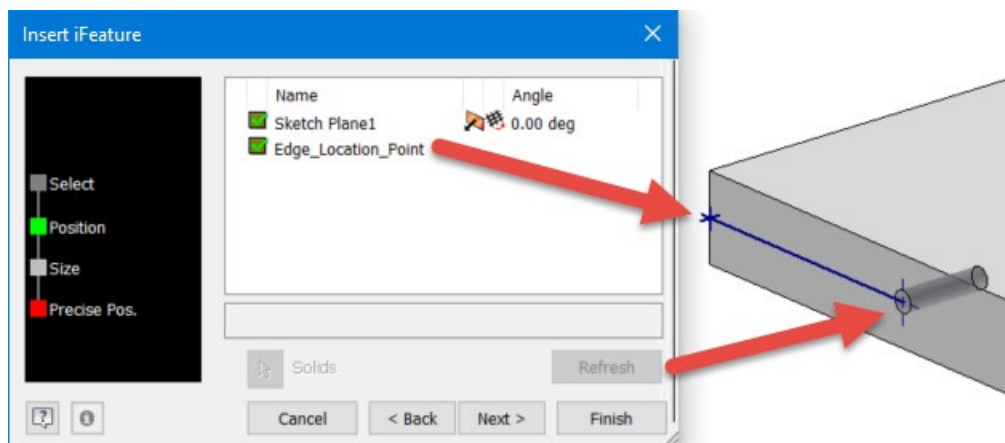
With the iFeature being published, naturally we should test it out. Placing this iFeature will not be appreciably different from some of the other iFeatures that we’ve placed, but I want to spend a little more time on this placement, since this is going to be a pattern of features versus a singular iFeature.

For this iFeature, please place it like you normally would. However, the end-user may not be aware that this particular iFeature should be placed at a midpoint (and they may

not read the prompts that closely 😊). Therefore, it is a good idea to click on the “Refresh” option to visualize the result. If the result is not good, the reselect the “Edge\_Location\_Point”, the prompt will be show below, and select the appropriate midpoint and “Refresh” one more time.



*RESULTS OF SELECTING A CORNER VERTEX VERSUS A MIDPOINT AFTER A “REFRESH” OF THE PLACEMENT (NOTE THE PROMPT BELOW THE SELECTIONS WHEN A SELECTION IS ACTIVE)*



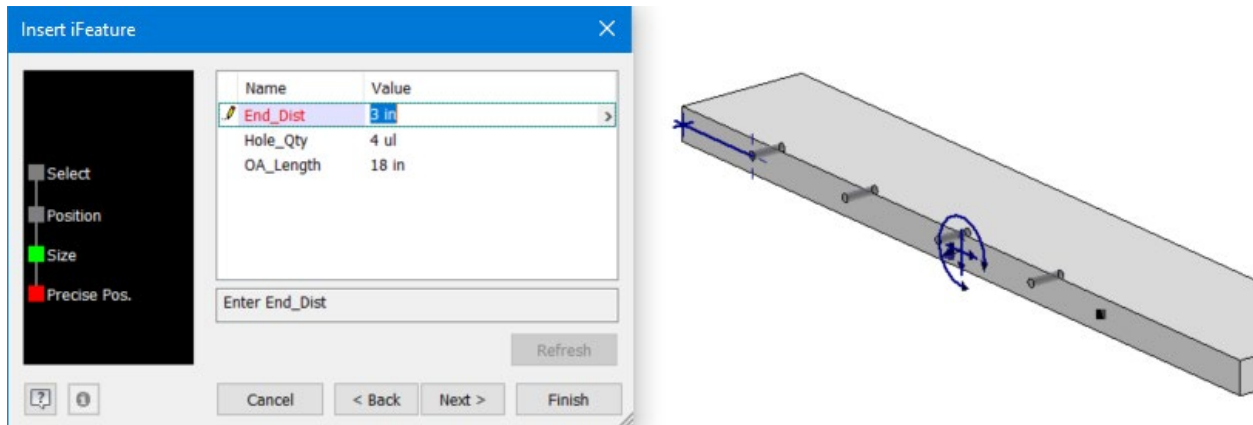
*RESULTS AFTER RESELECTING THE MIDPOINT OF THE LEFT VERTICAL EDGE AND PERFORMING A “REFRESH” OF THE PLACEMENT*

## Linking the pattern iFeature to host component parameters

Once the placement is completed and in a desirable state, we can move on to addressing the parameters. In the previous example, we actually linked parameters tying the K-Factor of the host component to the K-Factor within the iFeature. However, in this example, we'll take a little more time to understand this process, as this is particularly important to ensure that our pattern placement works correctly in every component in which this iFeature is placed.

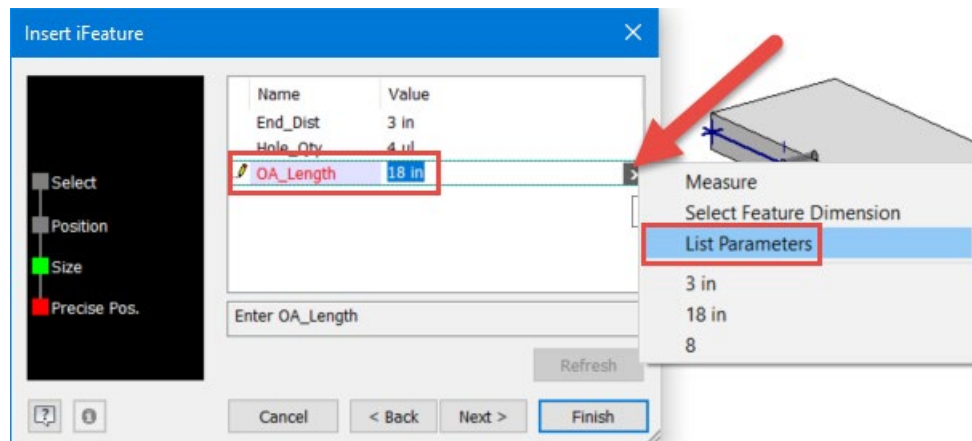
When editing the parameters for iFeatures, one can of course manually key in values, if those values do not violate any limits placed on them during the publishing process.

However, in this case, I want to tie the “OA\_Length” parameter to the length of the actual board in this design, as the “Refresh” clearly shows that the length of this board is greater than 18 inches.

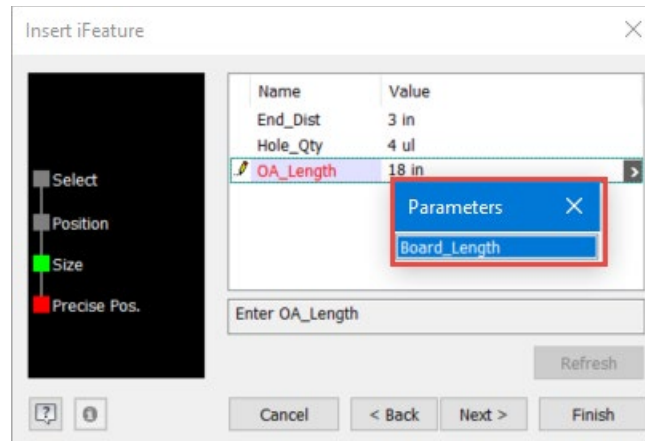


*PREVIEW OF THE HOLE PLACEMENT UTILIZING THE PUBLISHED PARAMETER VALUES*

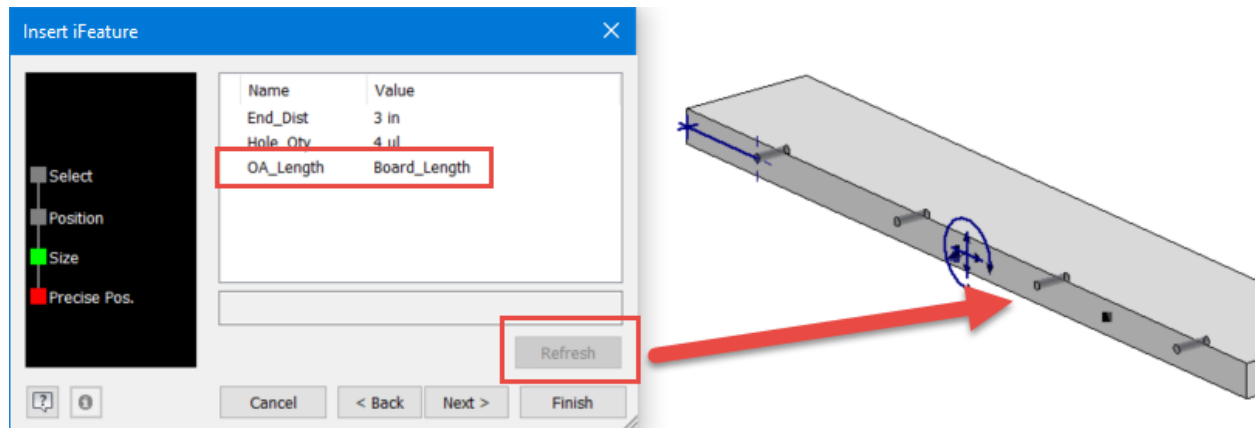
To link the parameters of an iFeature to the host component during placement is exactly the same as listing parameters in most other Inventor operations. See the images below.



*LEFT-CLICK ON THE ARROW ICON AT THE RIGHT-HAND SIDE OF THE DESIRED PARAMETER VALUE AND CHOOSE THE “LIST PARAMETERS” OPTION*



*SELECT THE DESIRED PARAMETER TO LINK BACK TO THE IFEATURE PARAMETER, IN THIS CASE THE "BOARD\_LENGTH"*



*RESULTS OF THE PARAMETER LINKING AFTER PERFORMING A "REFRESH" ON THE PLACEMENT*

This is a great way of linking the parameters during the placement, but what if we decide to add a host component control for the number of holes in the iFeature pattern. In that case we'll have to add a new parameter to the parameter table and then edit the iFeature to establish the link.



Parameters			
Parameter Name	Consumed by	Unit/Type	Equation
Model Parameters			
Board_Length	Hole_Pattern:1:OA_L...	in	24 in
d1	Extrusion1	in	4 in
d2	Extrusion1	deg	0.00 deg
d3	Sketch1	in	1 in
Hole_Pattern:1:End_Dist	Hole_Pattern:1	in	3 in
Hole_Pattern:1:OA_Length	Hole_Pattern:1	in	Board_Length
Hole_Pattern:1:Hole_Qty	Hole_Pattern:1	ul	4 ul
User Parameters			
Hole_Qty		ul	6 ul

☐ Link ☒ Immediate Update

GENERATE A NEW "HOLE\_QTY" PARAMETER, SET THE UNITS TO UNITLESS (UL) AND APPLY THE DESIRED QUANTITY

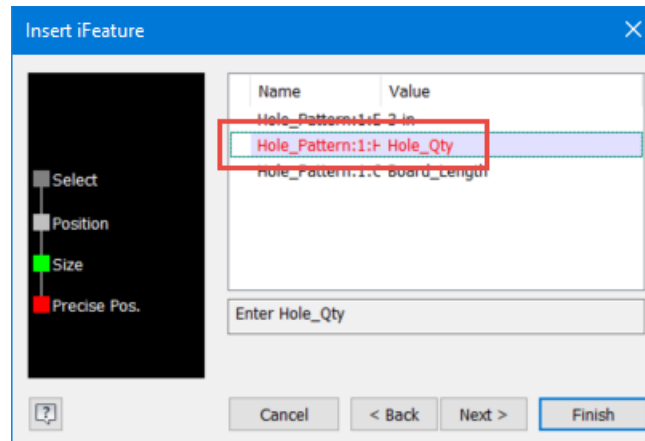
Once the desired "Hole\_Qty" parameter has been configured, the easiest way to link the parameter values is directly in the Parameter Table, as shown below. One could also edit the iFeature to accomplish this goal, but the Parameter Table linking will also show in the edit iFeature dialog box as well, so I recommend the easier route.

Parameters			
Parameter Name	Consumed by	Unit/Type	Equation
Model Parameters			
Board_Length	Hole_Pattern:1:OA_L...	in	24 in
d1	Extrusion1	in	4 in
d2	Extrusion1	deg	0.00 deg
d3	Sketch1	in	1 in
Hole_Pattern:1:End_Dist	Hole_Pattern:1	in	3 in
Hole_Pattern:1:OA_Length	Hole_Pattern:1	in	Board_Length
Hole_Pattern:1:Hole_Qty	Hole_Pattern:1	ul	Hole_Qty
User Parameters			
Hole_Qty	Hole_Pattern:1:Hole_...	ul	6 ul

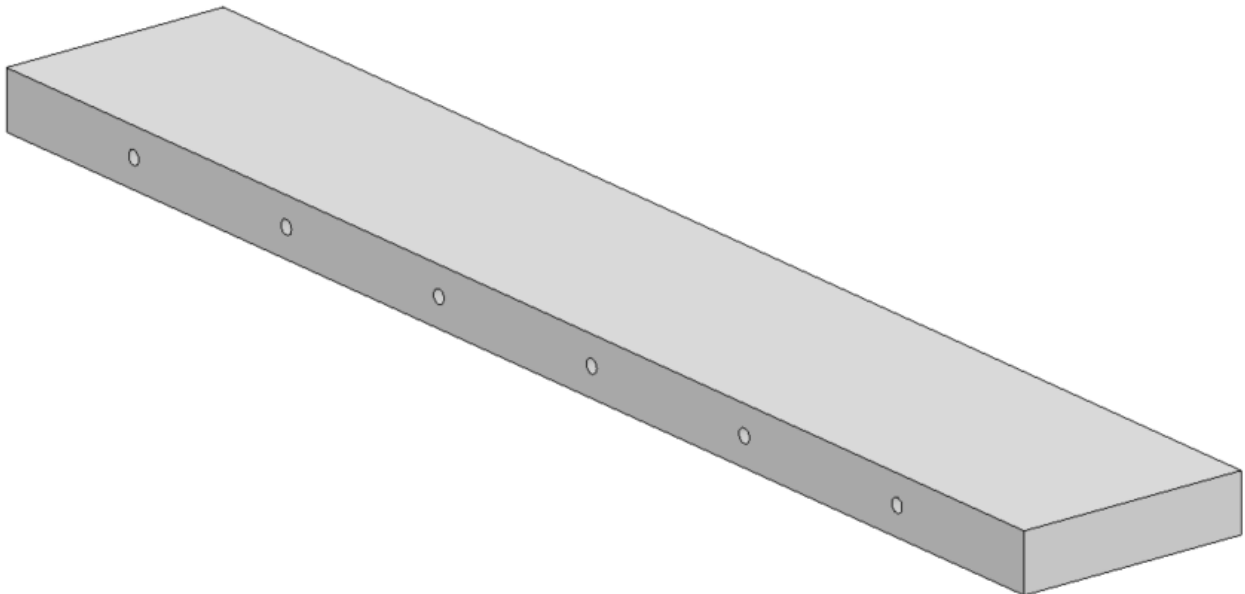
☐ Link ☒ Immediate Update

LINK THE IFEATURE "HOLE\_QTY" PARAMETER TO THE "HOLE\_QTY" OF THE HOST COMPONENT DIRECTLY IN THE PARAMETER TABLE





*EDITING THE IFEATURE SHOWS THAT THE PARAMETER LINK HAS ALREADY BEEN ESTABLISHED*



*FINAL RESULT SHOWING THE 6 HOLES EVENLY SPACED ALONG THE LONGEST EDGE OF THE BOARD*

## Conclusion

This concludes the class handout. As you've seen throughout this paper, there are many ways to think about authoring iFeatures and my goal has simply been to demonstrate some of the range of functionality these features can provide. Hopefully you find this helpful and if you have any questions or even more awesome examples, please share them!

# AUTODESK UNIVERSITY

## Acknowledgements

I cannot accomplish all of this on my own and I am blessed and thankful to recognize the following people who helped to make this course possible.

- God Himself for this wonderful opportunity to serve others and literally every breath I take!
- Rick VanWort, my current boss, and my former bosses for allowing me the time and space to explore these zany ideas
- All my customers who ask such interesting questions and provided the motivation to explore the particular examples that I shared in this class

## Appendix A: Weblinks to Video Demonstrations

Video demonstrating a single point placement iFeature

<https://www.youtube.com/watch?v=KoKpXsuHdLs>

Video demonstrating how to create a compound iFeature

<https://youtu.be/QUOjkJRbrnA>

Video demonstrating how to create a revolved emboss Sheet Metal Punch

<https://youtu.be/Zt3VT3e11I8>

Video demonstrating how to create an open-faced embossed Sheet Metal Punch (2 Sketches)

<https://youtu.be/OYsXI4TAyxE>

Video demonstrating how to create a pair of formed tabs as a Sheet Metal Punch

<https://youtu.be/g0LAKD1QPxQ>

Video demonstrating how to execute multiple extrude operations from the same instance of the Extrude command

<https://youtu.be/ZIVooe-uDtY>

Video demonstrating how to create an internal pattern iFeature

<https://www.youtube.com/watch?v=tPocxN5VSKc&t=2s>

## Appendix B: iLogic Examples

Text for “K\_Factor\_Extractor” iLogic Rule:

```
K_Factor = SheetMetal.ActiveKFactor
```

```
iLogicVb.UpdateWhenDone = True
```

## **Appendix C: iFeature Naming Considerations**

Renaming the iFeatures is a really important aspect of publishing user-friendly features, but there are some characters that are off limits. Here is a list of what you canNOT use when naming iFeatures:

- No spaces in the name
- No dashes “-“
- No punctuation (i.e. “,” “.” “;” etc.)
- No special characters (i.e. “~” “#” “%” etc.)
- No slashes “/” as in fractional values