

IM120928

Adding Intellect to Inventor with Intelligent Modeling

Jason Hunt
FS-Elliott Co., LLC

Kevin Smedley
SPC Mechanical

Learning Objectives

- Explore using parameters and the available functions in Inventor to build intelligent models
- Learn how to incorporate iLogic in your intelligent-modeling workflows
- Discover the power of using Adaptivity in your designs
- Learn how to use simplification to drive modular design

Description

Adding intelligence to your Inventor software designs is necessary when trying to speed up your design workflows. Inventor models don't have to be passive; they can have "intellect." Ask yourself these questions: Why should you always redo all the work every time you create a similar model? Do you find that you're creating similar models with only slight differences in size or configuration? Do you wish you could make your parts change when your layout changes? If you answered yes to any of these questions, this class is for you. We'll explore four different techniques to make your models work harder for you, covering the power of user parameters and functions, adaptive design, simplification, and iLogic. We'll learn how to use and apply these techniques to generate intelligent designs, improving the efficiency of design workflows.

Speaker(s)



Jason Hunt is the NPD Group Leader in the CAD Designer Group for FS-Elliott Co., LLC. FS-Elliott is a leading manufacturer of oil-free centrifugal air and gas compressors with sales, service, and manufacturing locations around the world. Based in Williamsville, NY, Jason provides lead design services to current NPD projects and helps drive current CAD standards and best practices among the NPD team. His experience involves 20+ years of compressor design, with an education background in Engineering from SUNY at Alfred, Industrial Engineering from SUNY Buffalo State and Business/Marketing from the McColl School of Business at Queens University of Charlotte.

jhunt@fs-elliott.com

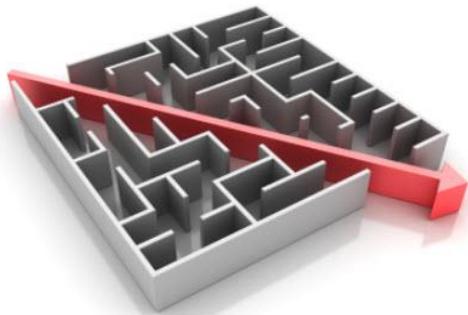


Kevin Smedley is the Cad Manager for SPC Mechanical headquartered in Wilson, NC. SPC Mechanical is an MEP Contractor in North Carolina and has four locations throughout the state. Applications for success are Revit, Navisworks, BIM, Sysque and Trimble. Kevin has over 28 years of CAD and Management experience with multiple Autodesk products in the manufacturing and AEC environments. His background includes CAD and design management, instructor/trainer, consultant, and implementation and support specialist. Additional areas are Engineering Systems Manager and CAD Administrator, product data management (PDM), product lifecycle management (PLM) with PMP and CIP. Kevin's specialties include CAD management, Inventor 3D parametric modeling application, Vault Professional software data management, Revit, BIM and the role of continued support specialist.

kevin.j.smedley@gmail.com

Contents

Learning Objectives	1
Speaker(s)	2
Class Overview	4
Utilizing Parameters to Create Intelligent Designs.....	5
Why should you use parameters when designing?	5
Basic Overview of Parameters	5
Equations and Parameters	8
Demonstration – Controlling a Stool with Parameters.....	11
Incorporating iLogic into your Intelligent Modeling Workflows.....	17
Why use iLogic to in Your Design?	17
Basic Overview of the 2018 Inventor iLogic Browser	17
Demonstration - Controlling a Stool Design with iLogic.....	20
Incorporating Adaptivity in your Designs	27
Why Use Adaptivity in Your Design?	27
Demonstration - Utilizing Adaptivity to Create Conceptual Stool Designs	28
Simplification and Modular Design	30
Why Create a Modular Design?	30
Planning Ahead	30
Demonstration – Driving Modular Design with Simplification	32
Conclusions	38



Simplification



Class Overview

Why does adding intelligence to your design matter? There are many different reasons that may drive you to utilize intelligent modeling. You may have a need to automate models and/or designs, because you may be dealing with repetitive, mundane tasks. Maybe you are finding that designers are making too many mistakes, when modeling common components. You may also be trying to build consistency of the modeling output to your CNC department to improve quality. Whatever the reason for adding intelligence to your design, you need to find a way to do it in Autodesk Inventor.



This class and paper will introduce a designer to methods in building intelligent models / designs. These methods are just an introduction to get you to think about how you can apply these methodologies to your current workflows.

Today, we are going to learn how to start out with the basics, using parameters to build intelligent flexible models and how using parameters can help with modifying a part. We will also discuss and demonstrate how you can use iLogic in conjunction with parameters to automate design variations. Next, we will show how to use Adaptivity in the design process. Our last topic will be about Simplification. We will explore how to use Simplification to drive leaner BOM assemblies and how this tool will assist in creating plug and play options for your designs.

We will learn how to use and apply these four techniques to generate intelligent designs, improving the efficiency of your design workflows.

Utilizing Parameters to Create Intelligent Designs

Why should you use parameters when designing?

Here are some questions to ask yourself, to see if you should utilize parameters to create intelligent designs.

- Is the work repetitive?
- Is there a need or desire to improve consistency?
- Do you want to improve efficiency?
- Do you need to create many models in a short amount of time?

If you answered yes to any of the questions above, then using parameters to create intelligent designs should be an important facet of your design process.

Basic Overview of Parameters

Parameters are an important facet of any model or design. Parameters store dimensional values, formulas, text information and more to drive your designs. They drive your sketches, extrusions, holes and many other features. Parameters are the driving force behind Autodesk Inventor's parametric modeling based design application.

The power of this application relies in the use of parameters in controlling the models, through the model and user parameters created by you. These parameters create an associativity between the model and your data, so that your data is intelligently used by this application.

Parameters are your “everyday” tool to drive your designs. Next, we will go over the two categories of parameters, with the first being model parameters and the last being user parameters.

In order to access these parameters, go to the “fx” parameter button on the Quick Access Toolbar or to the Parameters Button on the Manage Ribbon.

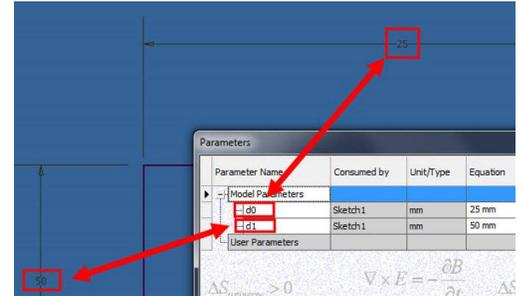


Every Parameter must have the following:

- A Parameter Name
- A Unit / Type
- An Equation or a Value

Model Parameters

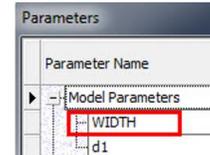
Model parameters are automatically created, as soon as you place a dimension in your sketch. Right away, these model parameters have an ID as a name. They start at d0 for the first one and then for every parameter created, after the first one, the ID changes in name sequentially. For example, d0, d1, d2, etc....



You have the ability to change the name of the parameter to something that makes sense to the user, for future use. In the example to the right, you can change the name of parameter by adding the name of the parameter “WIDTH” as part of the input.



By doing this, you have now renamed “d0” to “WIDTH”.



In this next example, d1 will have the parameter name entered as part of the equation. Take notice of the usage of the model parameter “WIDTH” in the formula.



The model parameter is then automatically re-named as “HEIGHT” in the parameters list.

Parameter Name	Consumed by	Unit/Type	Equation
Model Parameters			
WIDTH	HEIGHT, Sketch1	mm	25 mm
HEIGHT	Sketch1	mm	2 * WIDTH

Changing the names of the model parameters to something that makes sense to the user makes it easy for the user to re-use parameters in other dimensions and formulas.

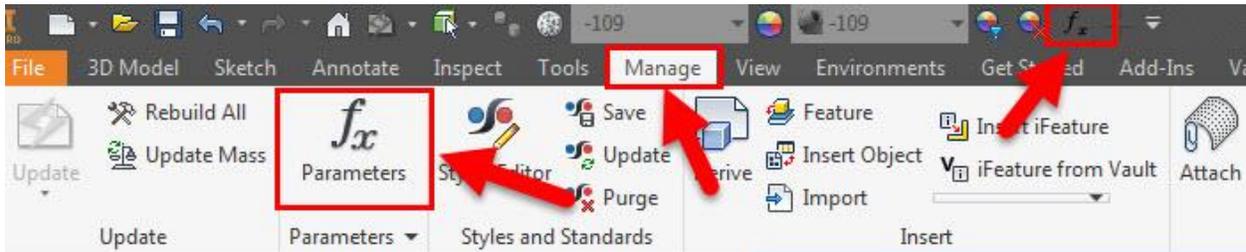
The main disadvantage in using model parameters is that they, more often than not, are accidentally deleted by users. This can wreak havoc on your models and design intent by the click of the mouse. The typical way this happens is by a user redefining a sketch and deleting one of the dimensions in the sketch. This in turn will delete any parameter or formula built into that model parameter. This is the main reason why I always recommend creating “User Parameters”.

User Parameters

The user manually adds “User” parameters and depending on your design intent are the most flexible. They are also stronger than model parameters and require you to go out of

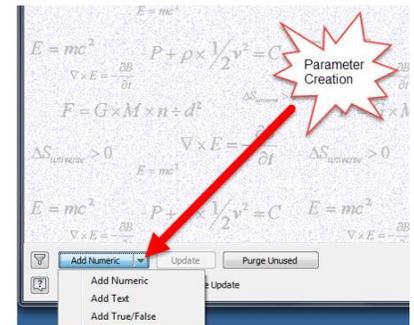
your way to delete them. There are choices in the type of user parameters to create and they are an excellent choice when writing equations.

In order to create a user parameter you have to go to the “fx” parameter button on the Quick Access Toolbar or to the Parameters Button on the Manage Ribbon.



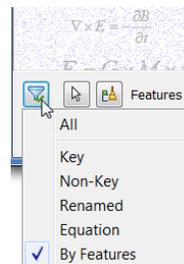
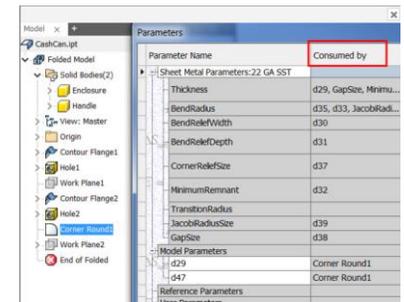
Once the parameter dialog box comes up, you need to click on the drop down arrow next to “Add Numeric” to see your three types of user parameters.

- **Numeric:** Used typically to drive geometry and in formulas / equations.
- **Text:** Some uses are to drive data on drawings or for assisting you in writing iLogic code.
- **True / False:** Typically used with iLogic



Parameters – What’s New in 2018

- A Consumed by column has been added to the parameters table to make it easy to identify where a parameter is used.
- By Features has been added to the parameter filter options.

Parameter Name	Consumed by
Thickness	d29, GapSize, Minimu...
BendRadius	d35, d33, JacobRad...
BendReliefWidth	d30
BendReliefDepth	d31
CornerReliefSize	d37
MinimumRemnant	d32
TransitionRadius	
JacobRadiusSize	d39
GapSize	d38
Model Parameters	d29
Reference Parameters	d47
User Parameters	

[2018 Overview of What's new](#)

[Productivity and Performance Enhancements](#)

Managing Parameter Display Within a Sketch

If you want to change how your numerical parameters are displayed within your sketch, you can change them. In order to do this you will need to right click in an open area of your sketch and then select “Dimension Display”. There are five options for you to be able to change your dimension display.

- **Value:** Shows the Dimension
- **Name:** Displays the dimension as the parameter name
- **Expression:** Displays the dimension as the expression
- **Tolerance:** Shows the Numerical value along with the tolerance for the dimension
- **Precise Value:** Shows the dimension value, ignoring all precision settings



Equations and Parameters

Please click on the below hyperlinks for more detailed information. There is a lot of good information in these two links.

[Equations and Parameters](#)

[Equations and Parameters \(Reference\)](#)

“You can use equations wherever you can enter a numeric value. For example, you can write equations in the Edit Dimensions dialog box, feature dialog boxes, and the Parameters dialog box. Equations can vary in complexity, and you can use them to calculate feature sizes, calculate assembly constraints offsets or angles, or simulate motion among several components. Equations can be simple or contain many algebraic operators, prefixes, and functions.

Algebraic operators

Autodesk Inventor supports the following algebraic operators:

+	addition
-	subtraction
%	floating point modulo
*	multiplication
/	division
^	power
(expression delimiter
)	expression delimiter
:	delimiter for multiargument functions Note: Comma was not used because it would conflict with floating point.
,	decimal delimiter ("." vs ";") in European countries.

Order of operations

Edit boxes use the algebraic order of operations, shown in the following table in descending precedence.

Operation	Symbol	Example
parenthesis	()	(sin(15 deg))
exponentiation	^	width^2
negation (unary subtraction)	-	(-2.00 + length)
multiplication or division	* or /	sin(pi/4 rad) or (0.5 * base * height)
addition or subtraction	+ or -	(-2.00 + height - 0.35 * base)

Functions

The following functions may be used in edit boxes.

Syntax	Return Type	Expected Types
cos(expr)	unitless	angle
sin(expr)	unitless	angle
tan(expr)	unitless	angle
acos(expr)	angle	unitless
asin(expr)	angle	unitless
atan(expr)	angle	unitless
cosh(expr)	unitless	angle
sinh(expr)	unitless	angle
tanh(expr)	unitless	angle
acosh(expr)	angle	unitless
asinh(expr)	angle	unitless
atanh(expr)	angle	unitless
cosh(expr)	unitless	angle
sinh(expr)	unitless	angle
tanh(expr)	unitless	angle
acosh(expr)	angle	unitless
asinh(expr)	angle	unitless
atanh(expr)	angle	unitless
sqrt(expr)	unit ^{1/2}	any
sign(expr)	unitless	any Return 0 if negative, 1 if positive
exp(expr)	unitless	any Return exponential power. For example, returns 2.688E43 for 100.
floor(expr)	ul	ul Next lowest whole number

ceil(expr)	ul	ul Next highest whole number
round(expr)	ul	ul Closest whole number
abs(expr)	any	any
max(expr1;expr2)	any	any
min(expr1;expr2)	any	any
ln(expr)	unitless	unitless
log(expr)	unitless	unitless
pow(expr1; expr2)	unit ^{expr2}	any, and unitless respectively Can construct unit valid equation that can go invalid "pow(3.0; d12)". Decimal powers rounds at eighth decimal place.
random()	unitless	unitless
isolate(expr;unit;unit)		Mechanical Desktop migration

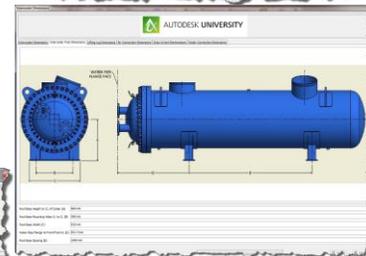
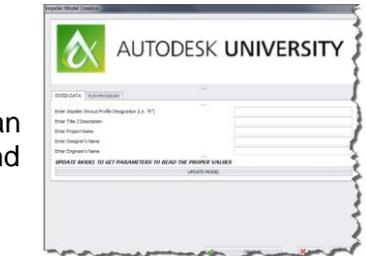
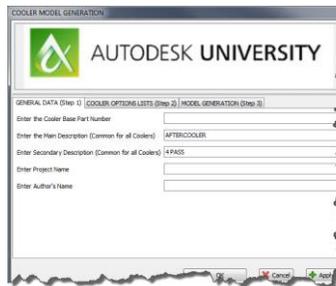
iLogic Forms:

What is a Form?

An iLogic form is essentially a user interface form. The form can be modal, add or edit controls on the iLogic browser, copy and paste a form, and use images and resize controls on forms.

The iLogic form can do the following:

- Connect to the parameters, properties and rules of the inventor part, assembly or drawing document.
- You can add and edit controls in the form.
- The appearance of the form can be modified.
 - Add controls into groups
 - Add images

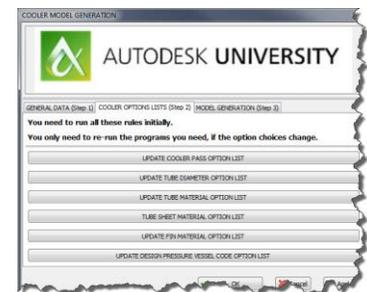


Why use forms?

Forms are a great tool to make it easier for the user to work on the Inventor file. If you created a master model to generate similar files repeatedly, then an iLogic form would be a great choice to use to ensure that the data is entered in an easily consistent manner.

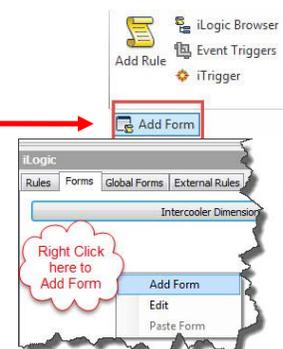
The form can assist the user in many ways.

- Running the rules
- Entering data
- Pictures can be added to illustrate clarity to what data needs to be entered and why
- Efficiency

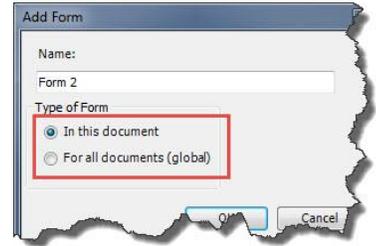


Creating iLogic Forms

- From the iLogic drop-down menu (Manage ► iLogic),
- Select "Add Form".
 - Alternate method is to right click in the iLogic Browser
 - If you create the form in the "Forms" Tab, the form will be internal to this document.
 - If you create the form in the "Global Forms" Tab, the form will be available in multiple documents.

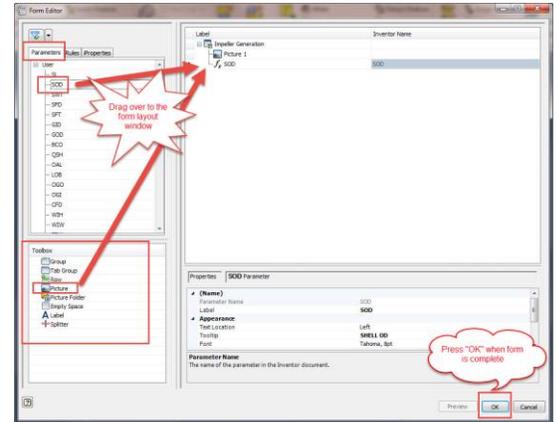


- After Selecting “Add Form”, enter a name for the form.
 - If you are creating the form from the iLogic Panel, you have a choice of making the form available in multiple documents. Select “For All Documents (Global)”.
- Design the layout of the form by dragging features from the Toolbox.
- Add controls by dragging from the Parameters, Rules, and iProperties tabs.
- To finish creating the form, click the “OK” button.



For more detailed information on creating forms, follow the Autodesk link below.

[iLogic Form Creation](#)



Demonstration – Controlling a Stool with Parameters

The purpose of this demonstration is to show how you can apply parameters and functions to control the design of a stool. This demonstration focuses on the usage of parameters for everyday usage.



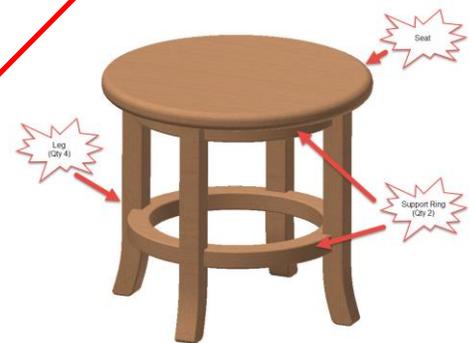
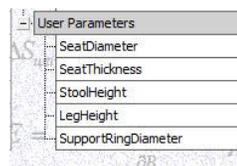
We will do the following:

- Create User Parameters to control the stool height and seat diameter.
 - Use a Trig Function to assist in the design
 - Push parameters to assembly members, from the assembly.
 - Create a form to manipulate your design on the fly

Creating the Stool User Parameters

1. The first step in this demo is to create the following “Numeric” User Parameters in the “Stool Parameters.ipt” file.

- SeatDiameter
- SeatThickness
- StoolHeight
- LegHeight
- SupportRingDiameter



These parameters will be used to drive the design, just as they mention in the names. They will control the Diameter of the seat and the thickness. The Height of the Stool and the Support Ring diameter will be controlled by a simple formulas.

- After creating the user parameters we need to populate the initial values and create formulas for the “LegHeight” and the “SupportRingDiameter” parameters. See below picture for values and formulas.

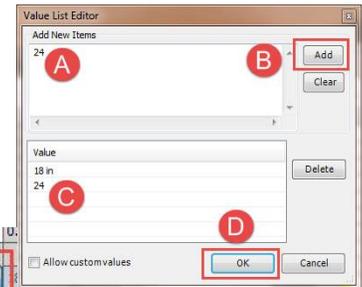
Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value	Tol.	Model Value	Key
SeatDiameter		in	18 in	18.000000		18.000000	
SeatThickness		in	1.25 in	1.250000		1.250000	
StoolHeight		in	20 in	20.000000		20.000000	
LegHeight		in	StoolHeight - SeatThickness	18.750000		18.750000	
SupportRingDiameter		in	SeatDiameter - 2 in	16.000000		16.000000	

Take note that the “LegHeight” parameter is a function of the “StoolHeight” Parameter minus the “SeatThickness” parameter and that the “SupportRingDiameter” parameter will always be 2 inches less than the “SeatDiameter” Parameter.

- Since we want to vary the height of the stool and the diameter of the seat, we will make the “SeatDiameter” and the “StoolHeight” parameters Multi-Value Parameters. To do this, you just need to right click on each one of the parameters and select “Make Multi Value”. A dialog box will pop up.



- Enter the Value “24” in the “Add New Items” section.
- Click the “Add” button.
- Take note that “24” is now added to the values section.
- Click the “OK” button.



You have now made “SeatDiameter” a multi-value parameter.

Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value	Tol.	Model Value	Key
SeatDiameter	SupportRingDiameter	in	18 in	18.000000		18.000000	
SeatThickness	LegHeight	in	1.25 in	1.250000		1.250000	
StoolHeight	LegHeight	in	24	24.000000		24.000000	

Do the same steps to the “StoolHeight” parameter, as above, but make the multi-values to be “20”, “28” and “36”.

Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value	Tol.	Model Value	Key
StoolHeight	LegHeight	in	20 in	20.000000		20.000000	
LegHeight		in	20 in	20.000000		20.000000	
SupportRingDiameter		in	28	28.000000		28.000000	
		in	36	36.000000		36.000000	

- The next step is to push these parameters to the appropriate components in the assembly.

- The first step in preparing to push the parameters to each file is to change the parameters to be “Exportable”. Go to the “fx” button, click on it so the parameter dialog box pops up, and select all the user parameters we created as “Export”.

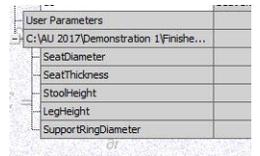
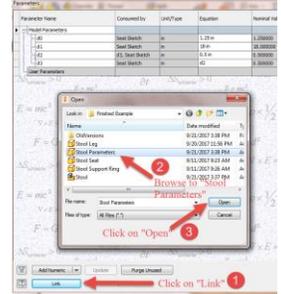
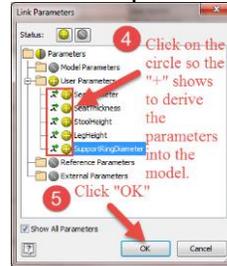
Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value	Tol.	Model Value	Key	Export Parameter
SeatDiameter	SupportRingDiam...	in	18 in	18.000000		18.000000		<input checked="" type="checkbox"/>
SeatThickness	LegHeight	in	1.25 in	1.250000		1.250000		<input checked="" type="checkbox"/>
StoolHeight	LegHeight	in	20 in	20.000000		20.000000		<input checked="" type="checkbox"/>
LegHeight		in	StoolHeight - SeatThickness	18.750000		18.750000		<input checked="" type="checkbox"/>
SupportRingDiameter		in	SeatDiameter - 2 in	16.000000		16.000000		<input checked="" type="checkbox"/>

B. Next, we will go to each component in the assembly, open them up and import the parameters from the “Stool Parameters.ipt” into each file.

- Stool Seat.ipt
- Stool Leg.ipt
- Stool Support Ring.ipt

Let’s open up Stool Seat.ipt and import the “Stool Parameters.ipt” parameters by going the “fx” button in the “Stool Seat.ipt” and doing the following:

1. Click on “Link”
2. Browse to “Stool Parameters” and select it
3. Push the “Open” button
4. Change the Desired Parameters to be derived
5. Click “OK”
6. Parameters are in the file. Now repeat the same steps for the “Stool Leg.ipt” and the “Stool Support Ring.ipt”.



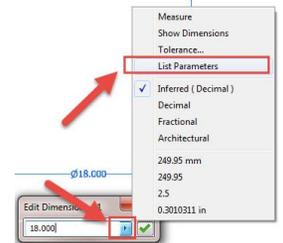
5. In this step, we are going to apply the parameters that we imported to each file to the sketch dimensions to drive the design parametrically.

A. The first part to apply the parameters to is the “Stool Seat.ipt”

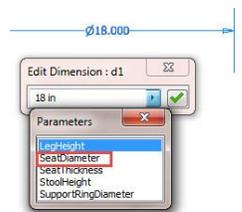
- First, we need to edit the “Seat Sketch”.



- Next, we will change the diameter dimension to read the parameter “SeatDiameter”, by editing the $\varnothing 18.000$ dimension and clicking on the arrow in the edit dimension dialog box, to select “List Parameters”.



- The “Parameters” dialog box will pop up. Select “SeatDiameter” and click on the green check box. You have now driven the seat diameter parametrically.
- Next, you will do the same steps to modifying the seat thickness, by editing the 1.25 dimension in the sketch. The only difference is to select the “SeatThickness” Parameter.



- Accept the sketch by clicking on the finish sketch button.

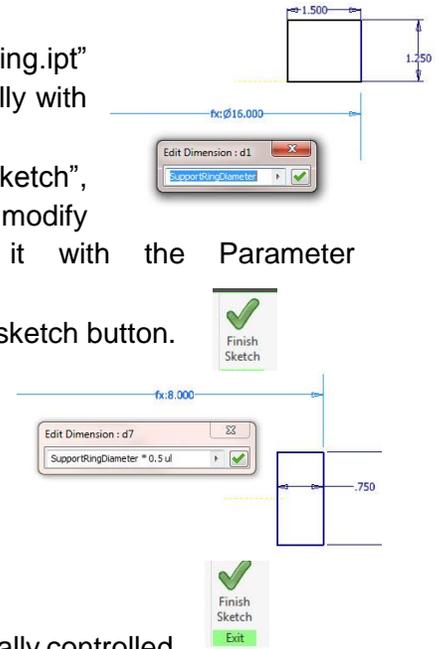


- Please note The “Consumed by” and “Equation” columns in the parameter dialog box. These columns show where the parameters are used, plus the equation used.

Parameter Name	Consumed by	Unit/Type	Equation	Normal value	Tol.	Model value	Key
Model Parameters							
d1	Seat Sketch	in	SeatThickness	1.250000	0.000000	1.250000	
d2	Seat Sketch	in	SeatDiameter	18.000000	0.000000	18.000000	
d3	Seat Sketch	in	d2	0.500000	0.000000	0.500000	
User Parameters							
d1		in		18.000000	0.000000	18.000000	
d2		in		1.250000	0.000000	1.250000	
d3		in		20.000000	0.000000	20.000000	
d4		in		18.750000	0.000000	18.750000	
d5		in		16.000000	0.000000	16.000000	

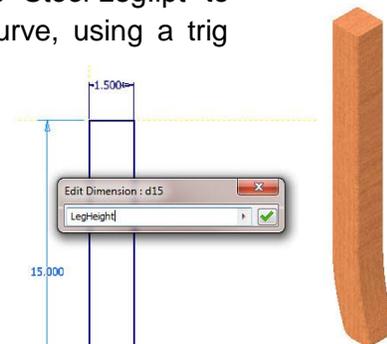
B. The next step is to open up the “Stool Support Ring.ipt” and control the support ring diameter parametrically with the parameter “SupportRingDiameter”.

- In the model tree edit the sketch “Ring Sketch”, found in the revolved feature “Ring”, and modify the $\varnothing 16.000$ dimension to control it with the Parameter “SupportRingDiameter”.
- Accept the sketch by clicking on the finish sketch button.
- In the model tree edit the sketch “Leg Notch Sketch”, found in the “Leg Notch” extrusion, and modify the shown dimension to control it with the formula “SupportRingDiameter * .5”.
- Accept the sketch by clicking on the finish sketch button.
- The “Stool Support Ring” is now parametrically controlled.



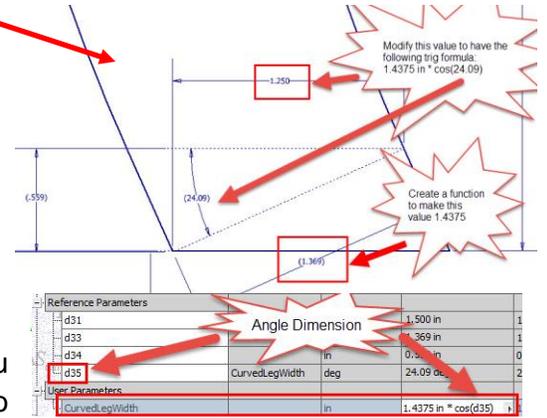
C. The last part to change parametrically is the “Stool Leg.ipt” to control the height and the width of the curve, using a trig function.

- In the model tree edit the sketch “Leg Sketch”, found in the “Leg” extrusion, and modify the 15.000 dimension to control it with the Parameter “LegHeight”.

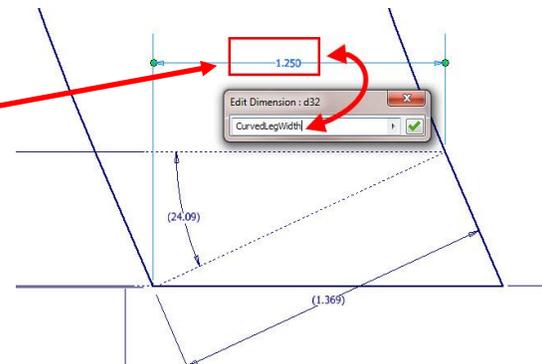


- The next feature we want to control is the width of the leg in the curved area of the same sketch using a trig function.

- Go to the parameter dialog box and edit the already existing parameter “CurvedLegWidth” to have the following formula “1.4375 * cos(d35)”. This is where you are solving for the leg value of the triangle to make the width equal to 1.4375 inches. I know you can do the width easier through constraints, but I wanted to show you how you could use a trig function to drive your design parametrically.



- In the sketch, modify the highlighted dimension in the picture to use the parameter “CurvedLegWidth”.

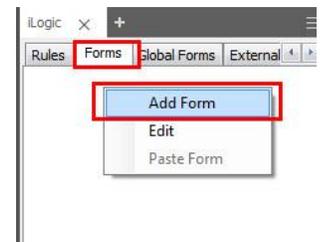


- Accept the sketch by clicking on the finish sketch button.
- The “Stool Leg“, now is parametrically controlled.

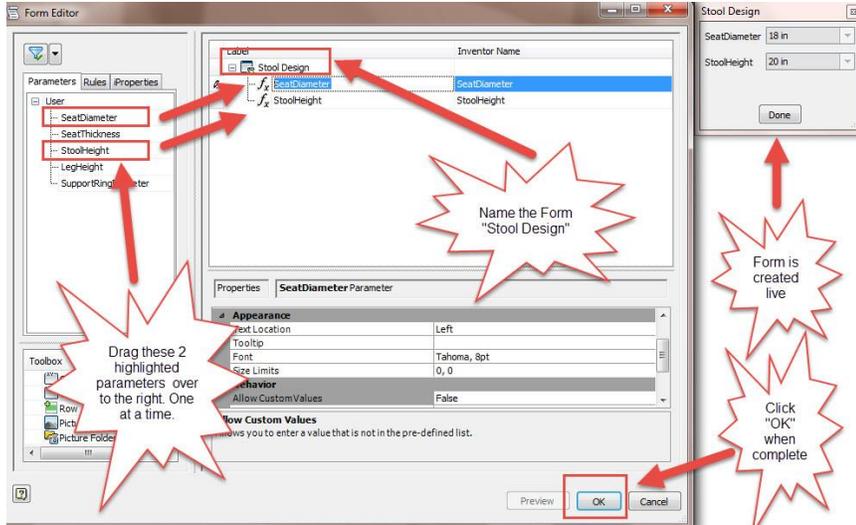


6. In this last step, we will add a form to the “Stool Parameters.ipt” to control the design.

- A. In the iLogic browser, go to the forms tab and right click in the “white area” and select “Add Form”



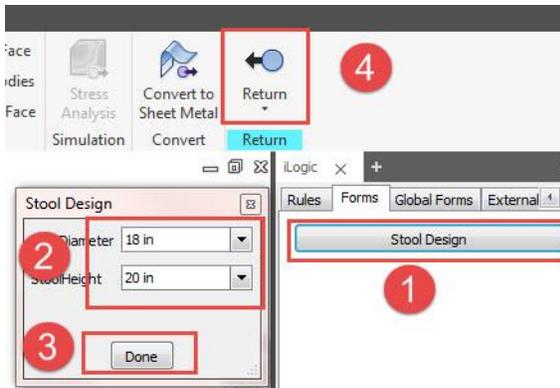
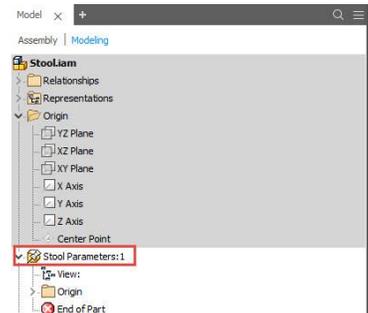
B. The form editor will pop up follow the instructions in the picture below, to create the form.



You now can use the form in “Stool Parameters.ipt” to change your design.

C. In order to access, this form go to “Stool.iam” and double click on “Stool Parameters” to modify the parameters within the assembly.

1. Go to the iLogic Browser “Forms” tab and select the “Stool Design” form.
2. The form will pop up. Modify the Seat Diameter and the Stool Height by making a choice from the drop down menus.
3. Push the “Done” button.
4. Push the “Return” button to see the stool change in size.



Incorporating iLogic into your Intelligent Modeling Workflows

Why use iLogic to in Your Design?

Here are some questions to ask yourself, to see if you should use iLogic in your design process.

- Is the work repetitive?
- Will automation save time, so a designer can focus on other tasks?
- Is there a need or desire to improve consistency?
- Do you want to improve efficiency?
- Does the data output need to be provided in a certain manner, every time?
- Do you need to create a lot of models in a short amount of time?

If you answered yes to any of the questions above, then using iLogic should be an important facet of your design process.

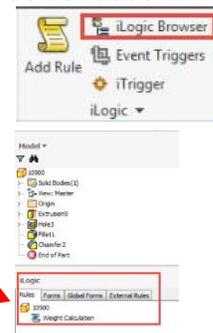
Basic Overview of the 2018 Inventor iLogic Browser

What is iLogic?

iLogic enables rule driven design that provides a simple way to capture and reuse your work. It allows the user to standardize and automate the design process.

iLogic allows you to become a coding expert without having to learn much actual code.

- In order to turn on your iLogic browser you need to go to the Manage Ribbon and to go to the iLogic panel and select “iLogic Browser”.
- The iLogic browser will either show under or above the model tree browser



What are iLogic Rules?

A rule is a small Visual Basic (VB.NET) program that can monitor and control other Inventor parameters, features, or components.

iLogic embeds rules as objects directly into a part, assembly and drawing documents. The rules determine and drive the design parameter and attribute values. By controlling these values, you can define the behavior of model attributes, features and components.

Knowledge is saved and stored directly in the documents, like the way in which geometric design elements are stored.

Internal Rules

iLogic Rules saved within a document are known as Internal Rules.

- To create an internal iLogic rule go to the iLogic browser and click on the “Rules” tab.
- Next right click in the open space and select “Add Rule” from the pop-up window.

When to use Internal Rules

The iLogic rule is only going to apply to one part and not globally.

- For example, if you have a part that can be any diameter, but can’t exceed the maximum or minimum diameter due to material availability. You could write an internal iLogic rule to flag the user.

An advantage to an internal rule is that it is copied with the part and will always remain with the part, no matter where that part is used.

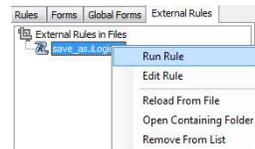
A big disadvantage is that if you need to edit or correct the code, you will have to track down every copy of that part to perform the edits. This can be time consuming.

It is up to the designer to decide if an internal iLogic rule is the best approach.

External Rules

External Rules are saved on your local or network drive.

- To create an external iLogic rule go to the iLogic browser and click on the “External Rules” tab.
- Next, right click in the open space and select “Add External Rule” from the pop-up window.
- iLogic will look for Rules in the following places:
 - The folder in which the current Inventor document is located.
 - The current Inventor Project Workspace folder.
 - The list of folders set in iLogic Configuration.
- Next, you will have to select the External Rule and run it.

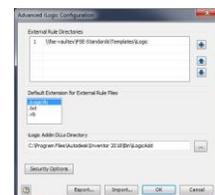


If you plan on sharing your Rules with others in your company, it is a good idea to add your company iLogic folder to the list of folders in the iLogic configuration setup.

To do this, go to “Tools > Options > iLogic Configuration”



A pop up window will appear so you can add your company iLogic Folder to the set up.



When to Use External Rules

External Rules are great to use when one wishes to apply the rule to many models.

- For example, if you have a company requirement that all panels will have set choices for material thickness, length, width, height and color. Then you could write an external iLogic rule to drive these choices through all the panels.

One advantage in this case is that any errors in your code can be fixed in the one source file and will automatically be applied, every time they are called in a document versus going into every model that has the rule embedded in it.

Tip: External rules are great for creating iLogic code 'Modules' that you can reuse for other tasks.

The only disadvantage is that you will need to remember to send your iLogic file with the document, if you want the iLogic Rule to be used elsewhere.

Best Practices for Writing iLogic Rules

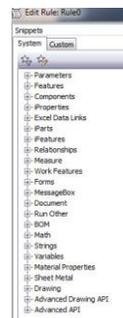
- Use comments in your code to make it easier to understand what your code is doing. This will help you and others down the road to understand how it works.
- Don't overdo it by overcomplicating your rule. Sometimes more rules are better than putting them all into one.
- Consider making your rules, so they can be reused in other projects. Why reinvent the wheel.
- When writing code, it is always good to be consistent in your methods.
- Did I mention to use comments?

iLogic Code Snippets

Code snippets provide the programmer shortcuts, for frequently used pieces of code. Using snippets allows the user to insert them into your code that you would normally have to type in manually. Using snippets also helps reduce the possibility of errors in your program, due to typographical errors.

You access the available snippets from the Snippets area of the Edit Rule dialog box. This area features two tabs:

- The System tab includes a set of predefined snippets, arranged by category.
 - In order to display the tool tip, hold the cursor over each snippet to display its function in more detail.
- The Custom tab allows you to add your own snippets, or create custom copies of System snippets.
- Favorite snippets
 - Favorites allow you to choose which snippets appear on the System tab. You can mark specific snippets as favorites, and then toggle the display of the list to show only those snippets marked as favorites.



Demonstration - Controlling a Stool Design with iLogic

The purpose of this demonstration is to show how you can utilize iLogic and parameters together to create a master assembly to generate an unlimited number of stool variations. This demonstration is a continuation of the first demonstration, but takes it quite a few steps further. We will focus more on the iLogic portion of the demonstration, as we already learned how to create and manage parameters in the first demonstration.



We will do the following:

- Create iLogic rules in each component in the “Stool” assembly to generate new models, based on the user inputs in the form.
- Create an iLogic Form to drive the design, via parameters and iLogic.
- Generate an iLogic rule in the Stool assembly to make and save the new stool design into a new folder a new assembly, with the new components.

Creating iLogic to Drive the Design

1. The first item to point out in this demo is that the user parameters have been created in the “Stool.iam” and in every one of the components in the assembly. We already learned how to create parameters in the first demo, no need to focus on the creation for this demo. The only need is to understand how they will be used in the iLogic code (to be explained later).

A. “Stool Seat.ipt” user parameters:



Parameter Name	Con	Unit/Type
Model Parameters		
User Parameters		
SeatDiameter	d1	in
SeatThickness	d0	in
Folder		Text
SeatPN		Text

B. “Stool Leg.ipt” user parameters:



Parameter Name	Con	Unit/Type
Model Parameters		
Reference Parameters		
User Parameters		
CurvedLegWidth	d32	in
LegHeight	d15	in
LegPN		Text
Folder		Text

C. “Stool Support Ring.ipt” user parameters:



Parameter Name	Con	Unit/Type
Model Parameters		
User Parameters		
SupportRingDiameter	d...	in
Folder		Text
SupportRingPN		Text

D. “Stool.iam” user parameters:



Parameter Name	Con	Unit/Type
Model Parameters		
User Parameters		
SeatDiameter	S...	in
SeatThickness	L...	in
StoolHeight	L...	in
LegHeight		in
SupportRingDiameter		in
PartNumber		Text
Folder		Text
SeatPN		Text
LegPN		Text
SupportRingPN		Text

2. The Next Step is to create an iLogic rule called “File Save As” in the “Stool Seat.ipt”. This rule will eventually be driven through the “Stool.iam” file to create a new seat model, based on user inputs. Let’s Get Started:

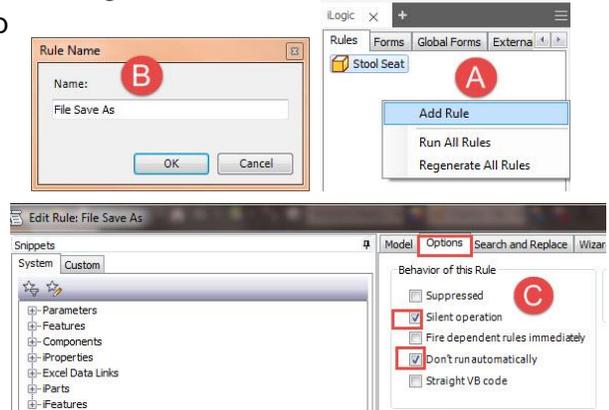
A. In the iLogic browser navigate to the “Rules” tab and right click in the white space and select “Add Rule” in the pop-up window.

B. Create the rule “File Save As”

C. Once the “Rule Dialog Window” pops up. Navigate to the “Options” tab and make sure “Silent Operation: and “Don’t run automatically” is checked. This ensures the rule does not run every time you update a parameter.

D. Enter the following iLogic code into the rule window as shown:

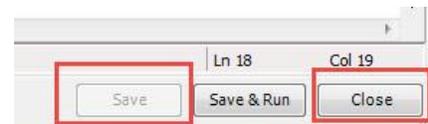
- This line of code creates an internal string parameter “FileToSave” that will be used to store the file name and path where to save the file.
- These two lines of code updates the model before saving.
- This line of code makes the iProperty “Part Number” equal to the “SeatPN” user parameter.
- This is the line of code is where we set the folder location to save to along with the new file name and store it in the string parameter “FileToSave”.
- The last line of code is where we are saving the new file to the new location



```

'----START OF iLogic
'----DEFINE VARIABLES FOR PROGRAM
Dim FileToSave As String 'DEFINITION TO SAVE MODEL TO A SPECIFIC FOLDER
'----Updating the Model
RuleParametersOutput ()
InventorVb.DocumentUpdate ()
'----Setting the Part Number iProperty to Match The Model Name
iProperties.Value("Project", "Part Number") = SeatPN
'----Save File to New PN
FileToSave = ThisDoc.Path & "\"+Folder+"\" + SeatPN + ".ipt"
ThisDoc.Document.SaveAs(FileToSave, True)
'----End OF iLogic
  
```

E. Save and close this rule. Be sure to save your model.



3. The next step is to add the same rule and code to each of other stool component models. Follow the same procedure as shown in step 2; just substitute the parameters as shown in the picture below.

Add the iLogic code to these parts:

- a. "Stool Leg.ipt"
- b. "Stool Support Ring.ipt"

```
'----START OF iLogic

'----DEFINE VARIABLES FOR PROGRAM
Dim FileToSave As String 'DEFINITION TO SAVE MODEL TO A SPECIFIC FOLDER

'----Updating the Model
RuleParametersOutput ()
InventorVb.DocumentUpdate ()

'----Setting the Part Number iProperty to Match The Model Name
iProperties.Value("Project", "Part Number") = SeatPN

'----Save File to New PN
FileToSave = ThisDoc.Path & "\"+Folder+"\\" + SeatPN + ".ipt"
ThisDoc.Document.SaveAs(FileToSave, True)

'----End OF iLogic
```

USE THE FOLLOWING PARAMETERS FOR THE LISTED PARTS:

1. Stool Leg.ipt = LegPN
2. Stool Leg.ipt = SupportRingPN

4. The next step is to create a rule called "Stool Design". Follow the same procedure as illustrated in steps 2A through 2C, except the rule name will be "Stool Design". This rule will essentially create a new stool model and save the model to a new location based on the user inputs entered in the iLogic form we will create after writing this code.



- A. The first part of this code we will define the following parameters and populate the Part Number iProperty.

```
'----START OF iLogic

'----DEFINE VARIABLES FOR PROGRAM
Dim FileToSave As String 'DEFINITION TO SAVE MODEL TO A SPECIFIC FOLDER
Dim FileToReplace As String 'DEFINITION TO REPLACE MODEL FROM A SPECIFIC FOLDER
iProperties.Value("Project", "Part Number") = PartNumber
```

- B. The next part of the rule is to push the user parameters values from the assembly we will enter through the iLogic form to each of the components in the assembly. See code below.

```
'----Reading the Parameters from the Assembly and Pushing them to Each Component
'----Push Parameters to the Seat Model
Parameter("Seat", "SeatDiameter") = SeatDiameter
Parameter("Seat", "Folder") = Folder
Parameter("Seat", "SeatPN") = SeatPN

'----Push Parameters to the Support Ring Model
Parameter("Stool Support Ring (Top)", "SupportRingDiameter") = SupportRingDiameter
Parameter("Stool Support Ring (Top)", "Folder") = Folder
Parameter("Stool Support Ring (Top)", "SupportRingPN") = SupportRingPN

'----Push Parameters to the Leg Model
Parameter("Leg1", "LegHeight") = LegHeight
Parameter("Leg1", "Folder") = Folder
Parameter("Leg1", "LegPN") = LegPN
```

- C. In steps 2 & 3, we wrote iLogic rules in each component found in this assembly. This part of the rule in the stool assembly is where we utilize these rules. What you are going to do is write the following iLogic code to run these rules in each of these components through the stool assembly. The following bulleted items is what happens in this step for each component.
- Update the model, based on the new model values entered in the user parameters. These new values were pushed to model in the previous step.
 - Save the file to the new name and new location.

```
'----Run "File Save As" Rules in Assembly Components to Generate New Models
iLogicVb.RunRule("Seat", "File Save As") '----Run Seat Model "File Save As" Rule
iLogicVb.RunRule("Leg1", "File Save As") '----Run Leg Model "File Save As" Rule
iLogicVb.RunRule("Stool Support Ring (Top)", "File Save As") '----Run Support Ring Model "File Save As" Rule
```

- D. In this next portion of the rule, we are going to write the code that will do the following in the “Stool.iam” model.
- The code will define to the correct folder path structure in the “FileToReplace” parameter to get to the new models.
 - The code will also do a component replace to put the new component models in the “Stool.iam” model.

```
'----Replace Components in Stool Assembly to new Models
FileToReplace = ThisDoc.Path & "\"+Folder+"\"
Component.Replace("Seat", FileToReplace + SeatPN + ".ipt", True)
Component.Replace("Leg1", FileToReplace + LegPN + ".ipt", True)
Component.Replace("Stool Support Ring (Top)", FileToReplace + SupportRingPN + ".ipt", True)
Component.Replace("Stool Support Ring (Bottom)", FileToReplace + SupportRingPN + ".ipt", True)

'----Define Path to Replace Model From
'----Replace Seat Model with New Model
'----Replace Leg Model with New Model
'----Replace Support Ring Model (Top) with New Model
'----Replace Support Ring Model (Bottom) with New Model
```

E. The code below will do the following:

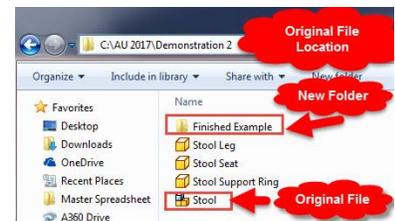
- The first portion updates the “Stool.iam” after replacing all the components. We need to do this before saving the assembly to a new name and a new location.
- The code will define to the correct folder path structure in the “FileToSave” parameter in order to save the “Stool.iam” to a new model name.
- The final part of the code will save the “Stool.iam” to a new name and location.

```
'----Updating the Model
RuleParametersOutput ()
InventorVb.DocumentUpdate ()

'----Save File To New PN
FileToSave = ThisDoc.Path & "\"+Folder+"\ " + PartNumber + ".iam"           'Define Path to Save to
ThisDoc.Document.SaveAs (FileToSave, True)
```

F. The code below will do the following:

- The first portion updates the “Stool.iam” after replacing all the components. We need to do this before saving the assembly to a new name and a new location.
- The code will define to the correct folder path structure in the “FileToSave” parameter in order to save the “Stool.iam” to a new model name. The path starts with the path to the “Stool.iam” and adds the subfolder within that folder to the new folder name.
- The final part of the code will save the “Stool.iam” to a new name and location.

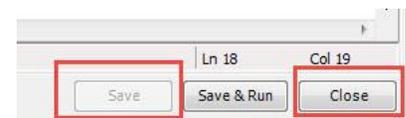


```
'----Restore Default Models
Component.Replace ("Seat", "Stool Seat.ipt", True) '----Replace Seat Model with Original Model
Component.Replace ("Leg1", "Stool Leg.ipt", True) '----Replace Leg Model with Original Model
Component.Replace ("Stool Support Ring (Top)", "Stool Support Ring.ipt", True) '----Replace Support Ring Model (Top) with Original Model
Component.Replace ("Stool Support Ring (Bottom)", "Stool Support Ring.ipt", True) '----Replace Support Ring Model (Bottom) with Original Model

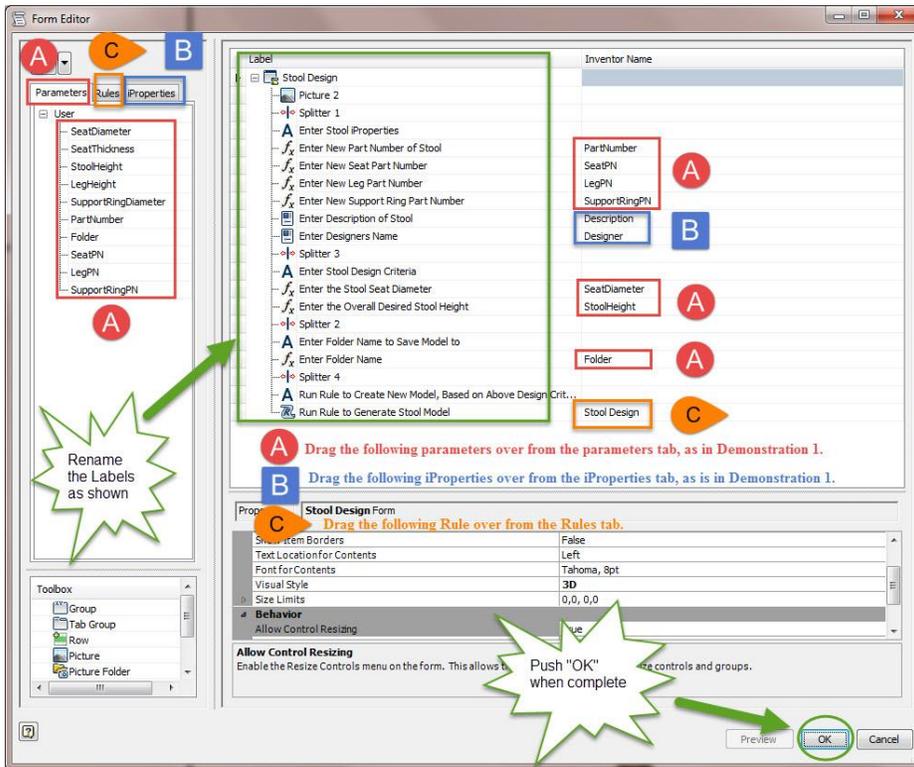
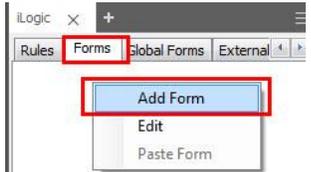
'----Updating the Model and Saving
RuleParametersOutput ()
InventorVb.DocumentUpdate ()
ThisDoc.Save

'----END OF iLogic
```

G. This concludes the writing of the code. Be sure to save and close this rule. The next step will be to create the iLogic Form to Run/Control the program.

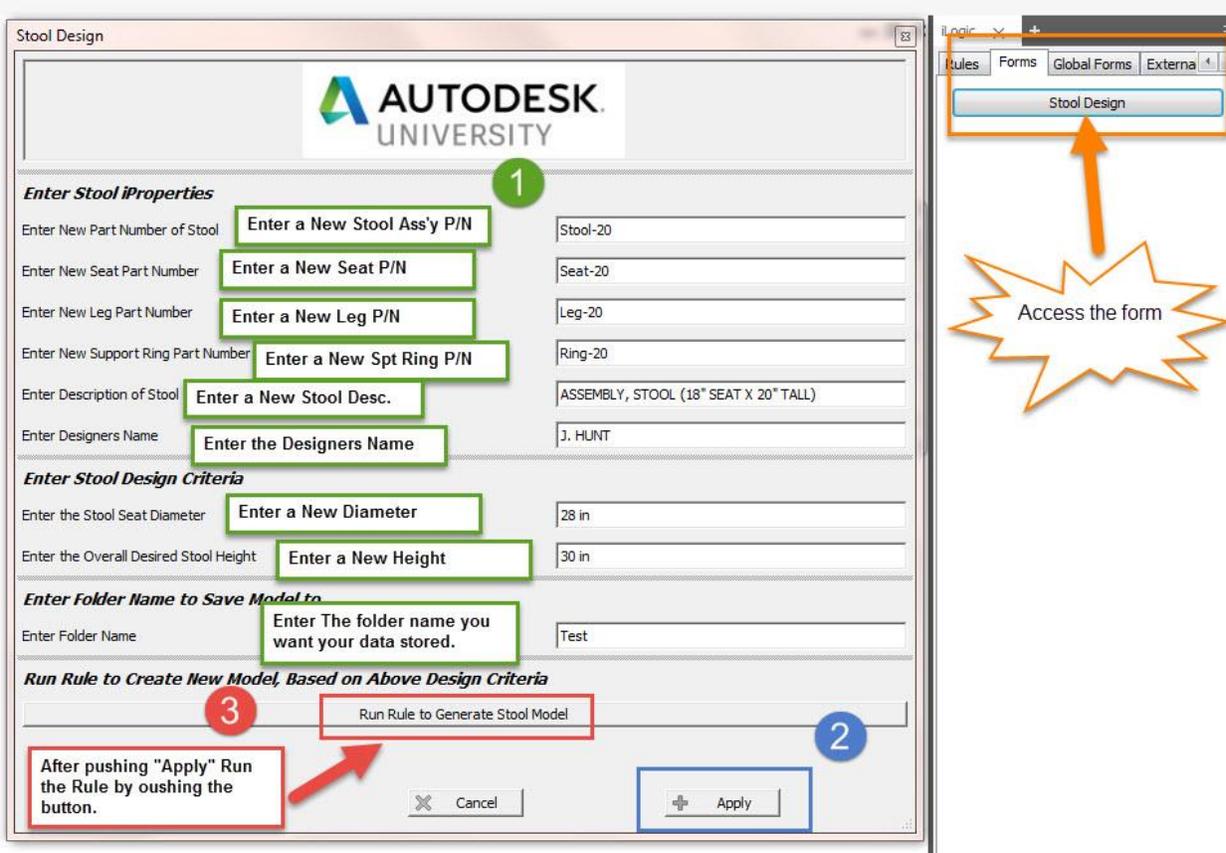


5. In this last step, we will add a form to the “Stool.iam” to control the design.
 - A. In the iLogic browser, go to the forms tab and right click in the “white area” and select “Add Form”
 - B. The form editor will pop up follow the instructions in the picture below, to create the form.



You now can use the form in “Stool Parameters.ipt” to change your design.

6. Now it is time to see the results. Access the form and play with it. Follow the steps in the below picture.



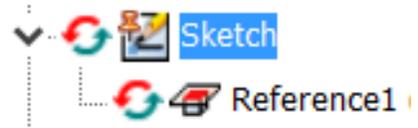
When complete, navigate to the folder you created to see the results. Remember, the location of the file starts where the original files started with a new sub folder create within the original folder.



Incorporating Adaptivity in your Designs

Why Use Adaptivity in Your Design?

What is **Adaptive Design**? Adaptive Design is a conceptual process that enables the creator to produce ideas and objects with more ease. The word “Adaptive” is an adjective that reflects adaptation or the ability to adapt. The word “Design” is a verb that describes action. For Engineers, Adaptive Design is a method that lays the basis for the making of every object or system while adapting to the changes of the surrounding objects and/or systems.



In the world of 3D Cad Designing (sketching, modeling, assembling and drawings), using adaptive design techniques can assist in producing quick ideas, objects and/or systems.

Autodesk Inventor 2018 and Adaptive Design  techniques are methods within workflows. Autodesk Inventor 2018 parametric modeling brings together an extensive number of techniques to accomplish most every type of creative industrial design. The two methods described in this handout are produced in a top-down design workflow. A top-down assembly level method for creating models is a highly recommended technique.

With Autodesk Inventor 2018, there are two main methodologies or techniques for Adaptive Designs:

- **Adaptive Cross-Part Sketch Geometry**
- **Adaptive Parts and Features**

Using adaptivity in Inventor is similar to turning a switch on and off. You turn on Adaptivity with either a right click or a check mark in the properties dialog. There are differences with the two methods of adaptivity and they can be characterized by referenced or under constrained.

You can find benefits with both of these techniques and reflect them through Adaptive and/or Flexible Sub-Assemblies. The techniques that you will learn can assist tremendously in producing components directly in relation to other components within an assembly and immediately reflect changes. Referencing existing models or sub-assemblies allows the creativity of the user to stay focused on the current idea.

As a user experiences with Autodesk Inventor 2018 modeling grows, they will learn the pluses and minuses to using Adaptivity. One downfall of Adaptivity is by using it excessively at any one time. If too much Adaptivity is attempted, there could be cyclical dependencies. This can result in a failed model that is trying to establish change. Adaptivity can also have outstanding benefits for quick assembly concepts and/or the consistency of an automated change process.

Demonstration - Utilizing Adaptivity to Create Conceptual Stool Designs

The purpose of this demonstration is to show how you can utilize adaptivity to generate some conceptual features from furniture company “X” to design a new feature to be sold as an option.

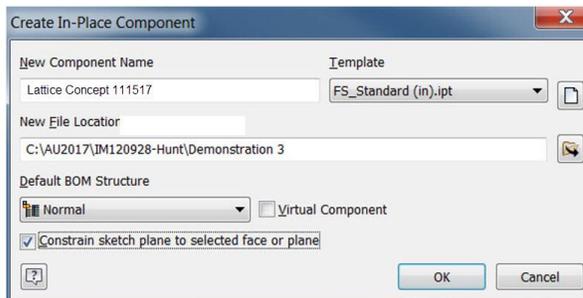
We will do the following:

- Utilize adaptivity to generate two conceptual stool design options.
- See how the power of adaptivity can be used to generate conceptual options.

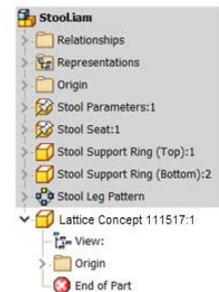
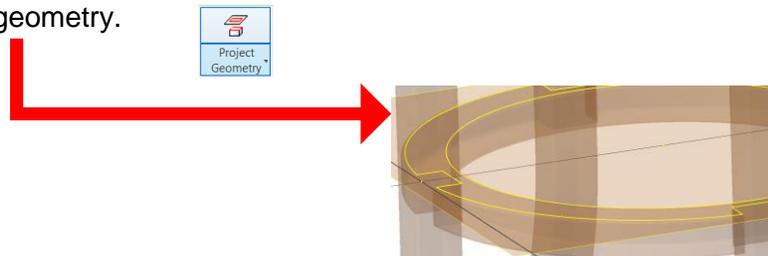


Creating Stool Conceptual options

1. Open “Stool.iam”
 - This contains five essential pieces and an overall parametric design that can be used when creating these concepts.
2. Create a new part within the “Stool.iam”.



3. Activate the Lattice model within the “Stool.iam” file and create a 2D sketch.
 - This will create Adaptive Cross-Part Sketch Geometry
 - Choose the bottom support ring to create the sketch plane and project the geometry.



4. We will create our first conceptual option sketch geometry using our adaptive capabilities within Inventor to create a lattice feature (See **Figure Concept I**).
5. Next, we will create a second concept (See **Figure Concept II**) following very similar steps, as shown in steps 1-4.
6. The final concept that shows both conceptual options in the “Stool.iam” model is shown in **Figure Concept I & II**.



Figure Concept I



Figure Concept II

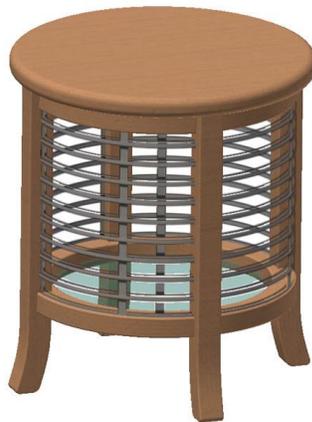


Figure Concept I & II

Simplification and Modular Design

There are many different methodologies used in defining the design process in the manufacturing realm. The following methodology will utilize the Simplification tool to simplify a compressor design; while at the same time streamlining the design process allowing multiple designers to collaborate on the design and work in an efficient way. We are going to use the Simplification tool to simplify the design process.



Why Create a Modular Design?

Questions to ask yourself about your Current Design Process:

- Do you have multiple Designers / Engineers Working on The same Project?
- Does your project entail creating multiple design options?
- Do you find it difficult to handle Design Scope Creep in an effective manner?
- Is speed to market crucial to meeting your goals?

If you answered yes to any of the questions above then creating smaller assemblies that are more manageable, should be an important facet of your design process. This can be accomplished by utilizing the Simplification tool, found in Autodesk Inventor.

The simplified model can be utilized in the following ways:

- The simplify model can be used to make a smaller more manageable file size to pass on to other groups within your Company. (i.e. Manufacturing, Projects Engineering & Sustaining Engineering).
- The Simplify model can also be used to assist in design. (*Our Main Focus for Today*)
- Used to create smaller assemblies that are more manageable, allowing more designers to collaborate on a project without stepping over each other, hindering efficient design.
- Used to generate General Layout drawings.

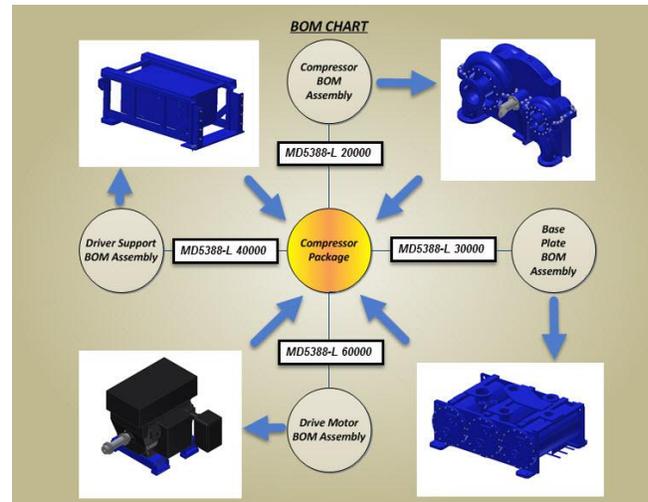
Planning Ahead

The design process just doesn't happen on its own. No matter the process used, there should always be a plan to determine the best direction to move forward with. The workflow process shown in this demonstration is based on 3D modeling practices and was created to assist in generating 3D geometry to populate the Bill of Materials (BOM), not 2D drawings.

Creating a Work Flow Diagram

Planning is the best use of an Engineer's or Designer's time. This in turn is crucial to streamlining the design process. A Work flow diagram is an excellent tool to help plan out the design.

This diagram either can be on paper or created digitally. No matter the method used, this document will assist you in simplifying how the 3D files are located and used within your BOMs. Thus, resulting in a streamlined process helping you to achieve maximum speed to market, which we all now is a very important in any design project.



Why a Workflow?

The methodology shown, utilized Visio to assist in creating the design workflow and we utilized this as a tool to simplify the 3D CAD data and to work with the overall "Simplification Strategy". The following bullet points listed below highlight criteria for this Simplification Strategy:

- Make it easier for Multiple Designers to work on the same project.
- Don't let your engineering 3D CAD Assemblies, be driven by the Manufacturing Assembly Drawing Requirement. Instead, create methods to support Manufacturing's requirements, without penalizing engineering 3D CAD structure.
- Create more flexibility in the Process to handle design options, Assembly Drawings and future design changes. (i.e. Plug and Play Options or Bolt-on Options)
- Speed to Market.
- Capability to drive BOMs from 3D CAD data (Autodesk Inventor).
- The ability to create a BOM that is accurate and easily exported.

The overall goal of this methodology is to make 3D CAD processes more efficient and better documented for consistency throughout the company. Thus, minimizing the amount of documentation and changes created outside of the Engineering / CAD departments; allowing designers to be more efficient and consistent. The ability to generate accurate BOMs from the 3D design data helps drive better business processes.

How the Workflow Diagram Drives the 3D Design

The workflow diagram is the roadmap for your project and is dependent on your company's workflows and 3D processes. This diagram will require significant thought, on how it affects the lifecycle of the CAD data, sustaining, and other departments that may be involved.

Benefits

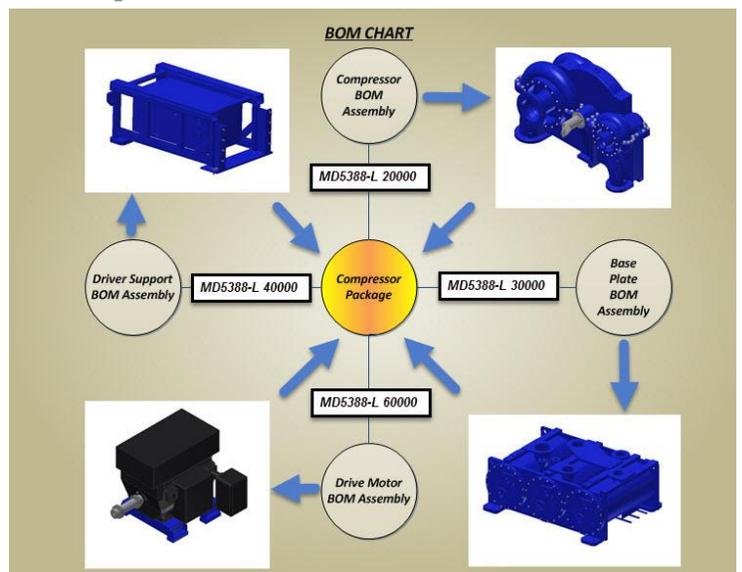
- Streamlined design Process
- Better Communication with:
 - Designers and Engineers
 - Departments outside of the Design Process
 - Customers
- Ability to work efficiently with:
 - Multiple CAD Designers
 - Multiple Engineers
 - Design teams
 - People Outside the Design Team
 - Vendors
 - Customers (Internal & External)
- The ability to handle scope creep in an effective manner

In summary, the workflow diagram is a game changer in processing designs in CAD and beyond. This document is visible and accessible to the whole team and company. Planning ahead to create these workflow diagrams are proven factors that help drive that requirement for speed to market for designs.

Demonstration - Driving Modular Design with Simplification

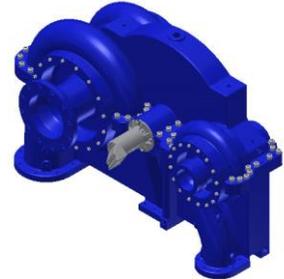
In this last demonstration, we are going to make a quick modular design using the simplification tool. We already have reviewed the design requirements and have created a workflow diagram showing our logical BOM assemblies.

- Compressor BOM Assembly
- Base Plate BOM Assembly
- Drive Motor BOM Assembly
- Driver Support BOM Assembly



Creating the Modular Design

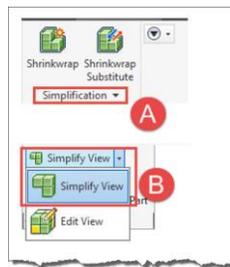
1. The first step is to figure out where to start. In this demonstration, we are going to start by opening up the compressor assembly (MD5388-L 20000.iam). This BOM is what is going to drive the “Common Coordinate System” throughout the design.



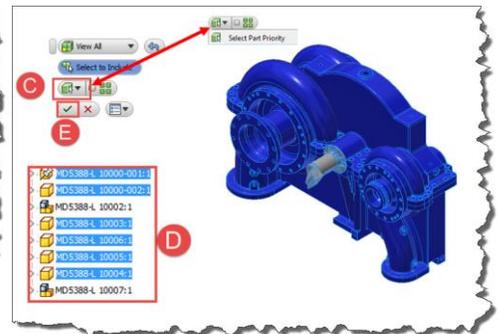
The reasons why this is important is as follows:

- All of your CAD data will be based on where you start.
- A common coordinate system will be used throughout the BOM assemblies.
- Helps drive consistency throughout the design.

- A. Go to the Assemble Ribbon in the “MD5388-L 20000.iam” file, go to the “Simplification” panel, and select the “Simplification” drop down arrow.



- B. Choose “Simplify View” to select the components to create a new simplified representation.

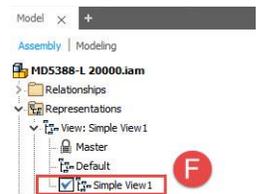


- C. The Simplification dialog box tool will pop up. Change the selection priority to “Select Part Priority”.

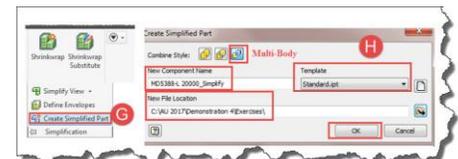
- D. Select all the components in the boxed in area of the step D in the picture, even the non-highlighted ones. Take note that the file edges are now highlighted in a light blue color.

- E. Accept your selections by pushing the green check button.

- F. You have now created a view representation called “Simple View1”. It is now wise to save the file to preserve the view representation.



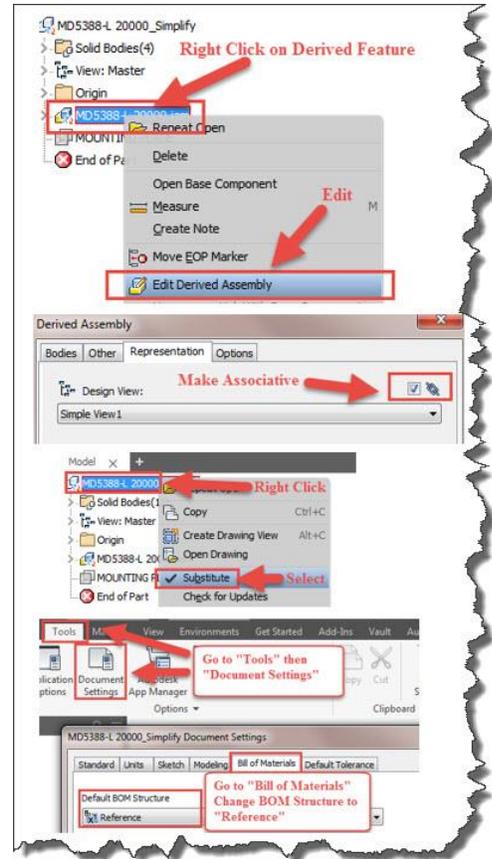
- G. The next step is to create simplify model to copy the desired geometry and copy the common coordinate system used throughout the design. Go to the “Simplification” panel and select “Create Simplified Part”.



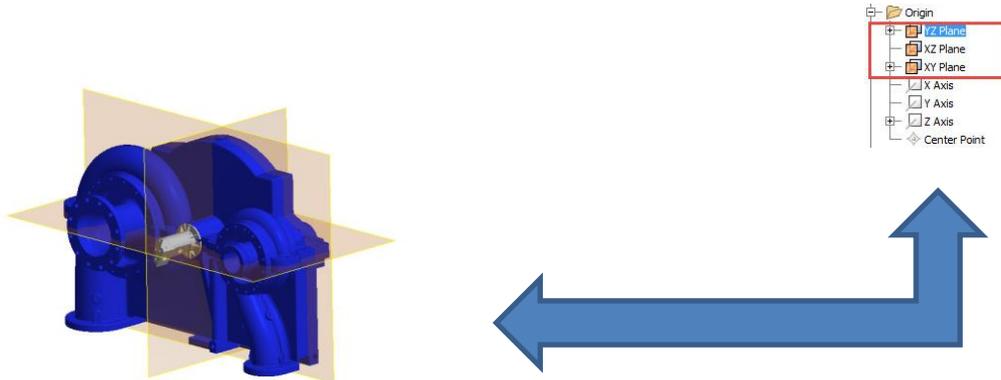
- H. The “Create Simplified Part” window appears. Do the Following:
 - Change the “Combine Style” to Multi-Body.
 - Change the Component Name to be “MD5388-L 20000_Simplify”.
 - Make the template be “Standard.ipj”.
 - Change your new file location to be where you want the file stored.
 - Push the “OK” button.

I. Now that the simplified part “MD5388-L 20000_Simplify.ipt” has been created, we need to change a few settings to make sure the simplify part will work as intended.

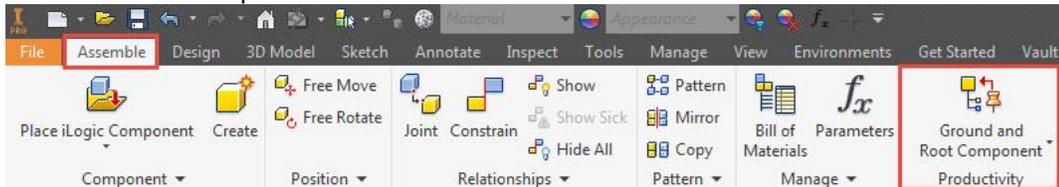
- In the model tree, “Right Click” on the derived feature to edit the derived assembly.
- The derived assembly pop-up window will appear. Go to the “Representation” and select the box (chain link icon) to make the derived feature associative to the parent assembly. Select “OK” to accept your selections
- Next, we are going to turn off automatic updates of the derived part. This will keep your files from pinging the parent assembly all the time, thus speeding up your workflows. “Right Click” on the file in the model tree and select “Substitute”.
- To check for updates down the road, select “Check for Updates” below “Substitute”
- The last step is to change the BOM settings to reference. This is done because you do not want to generate requirements in your BOM for a file that is used to drive your design/s. To do this you go the “Tools” Ribbon and select the “Document Settings” Panel. Once the Document settings window pops up go to the “Bill of Materials” tab and change the BOM structure to “Reference”. Push the “OK” button to preserve your selections.
- Save “MD5388-L 20000_Simplify.ipt”



J. A result of creating the simplify model of the compressor assembly, you also created a copy of the Default coordinate system of the assembly that will drive all other BOM assemblies.

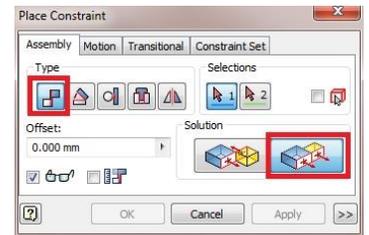
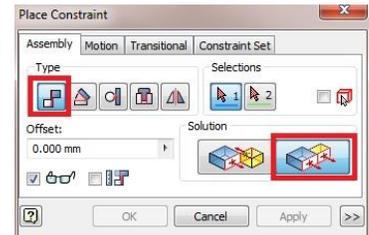
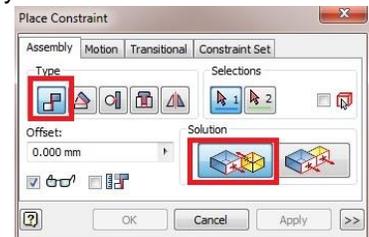


2. Open file MD5388-L 30000.iam
 - A. Place MD5388-L 20000_Simplify.ipt anywhere in assembly MD5388-L 30000.iam by clicking in inventor window and then pressing the escape Key.
 - B. In the browser select MD5388-L 20000_Simplify:1;
 - C. Expand the productivity panel in the assemble ribbon and scroll down and select “Ground and Root Component”.

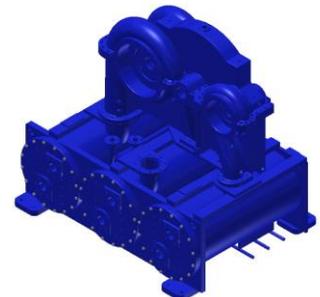


3. Constrain the following component (**MD5388-L 30001:1**) in the assembly as follows :

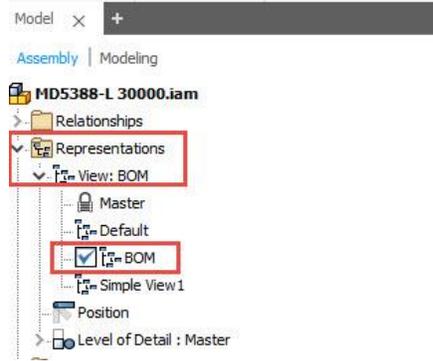
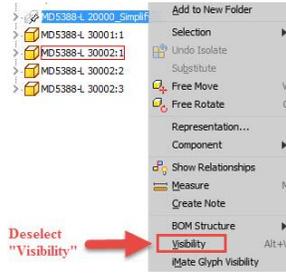
- A. Create a mate type constraint
 - i. The solution will be mate
 - ii. In the browser, expand “MD5388-L 30001:1” and select “MOUNTING PLANE” for selection one
 - iii. In the browser, expand “MD5388-L 20000_Simplify:1” and select “MOUNTING PLANE” for selection two
 - iv. Click apply
- B. Create a mate type constraint
 - i. The solution will be flush
 - ii. In the browser, expand “MD5388-L 30001:1” and select “CENTER PLANE” for selection one
 - iii. In the browser, expand “MD5388-L 20000_Simplify:1”
 - iv. In the browser, expand the origin folder and select “XY Plane” for selection two
 - v. Click apply
- C. Create a mate type constraint
 - i. The solution will be flush
 - ii. In the browser, expand “MD5388-L 30001:1” and select “AXIS CENTER PLANE” for selection one
 - iii. In the browser, expand “MD5388-L 20000_Simplify:1”
 - iv. In the browser, expand the origin folder and select “YZ Plane” for selection two
 - v. Select OK



- D. This assembly now shares the same “Common Default Coordinate System” as assembly MD5388-L 20000.iam. This is a result of the parent child relationship.
 - i. MD5388-L 20000.iam is the parent
 - ii. MD5388-L 30000.iam is the child



4. In the browser, under “Representations”, right click on “View” and a pop-up window will appear.
5. Select “New” to create a new View Representations and rename it to “BOM”. We are doing this so when assembling these files into a top level assembly, we don’t see the reference Simplify parts.
6. Save the file
7. In the browser right click on the simplify model MD5388-L 20000_Simplify and turn off the visibility.
8. Save the file
9. Close the file
10. Open files MD5388-L 40000.iam & MD5388-L 60000.iam.



- A. Go to the browser and take note that these assemblies already have the simplify models placed at the default origin (Common Coordinate system) in the model trees and all components are constrained to the simplify models.
11. Close files MD5388-L 40000.iam and MD5388-L 60000.iam
12. Open file MD5388-L 70000.iam from
 - A. You will be creating the compressor package assembly
13. Place the following files anywhere in the assembly by clicking in the window and then pressing the escape key for each file.

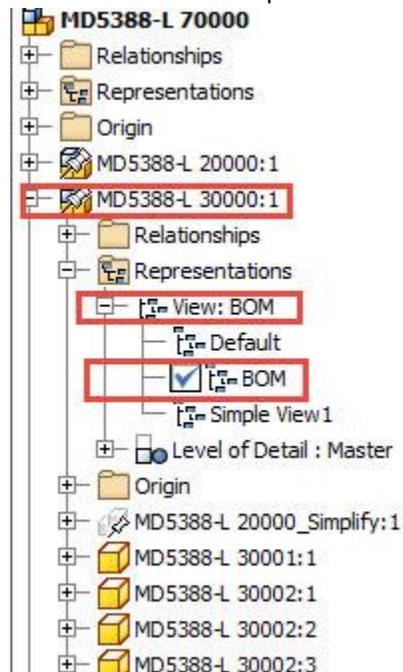
- MD5388-L 20000.iam
- MD5388-L 30000.iam
- MD5388-L 40000.iam
- MD5388-L 60000.iam

- A. In the browser expand the file name and activate the “BOM” view representation for the files by do the following:

File Names:

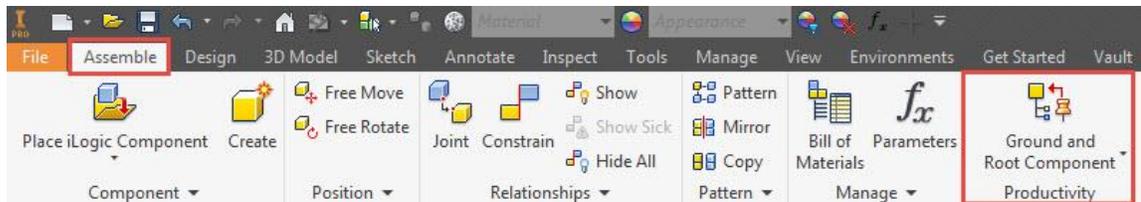
- MD5388-L 30000.iam
- MD5388-L 40000.iam
- MD5388-L 60000.iam

- In the browser, expand the representation folder in each part.
- Find the BOM view representation and right click on it.
- A popup window will appear, select “Activate”
 - i. MD5388-L 30000.iam
 - ii. MD5388-L 40000.iam
 - iii. MD5388-L 60000.iam

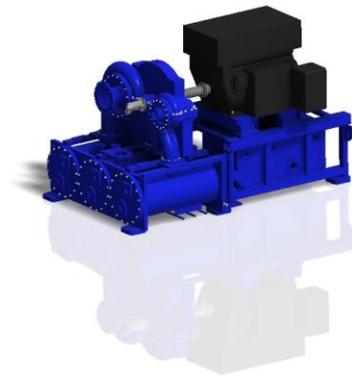


14. In the browser select the following files and “Ground and Root” the components one by one: (Expand the productivity panel in the assemble ribbon and scroll down and select “Ground and Root Component”).

- MD5388-L 20000:1
- MD5388-L 30000:1
- MD5388-L 40000:1
- MD5388-L 60000:1



15. You have just created the compressor package assembly with a common coordinate system
16. Save the file
17. Close all Files



How the Workflow Diagram Drives the 3D Design

We just created a modular design that is easily changed to reflect different design options. These options are easily replaced using this methodology, due to the common coordinate system being created and replicated into each BOM module. These design options can be as follows, but not limited to the following choices.

- Paint
- Code Requirements
- Different Motors
- Support Frame variations
- And more.....

The key to components to this methodology is determining what is going to be the **Common Coordinate System (CCS)** for your design and the utilization of these simplifies to drive the **CCS** throughout your design, while also using the “Ground and Root Component” tool in inventor.

Conclusions

There are a number of ways to create intelligent models within Inventor 2018. The key to deciding which method is best for your company is to understand your objectives. Each method has its own driving factor/s to push you towards using each method. I like to break it down as follows:

- Day to Day Tasks – Use simple parameters to drive your designs intelligently
- Master models / templates – Utilize iLogic & parameters to drive and create your designs
- Conceptual work – Use adaptivity to prove out conceptual designs or options
- Simplification – Modular design for Design Projects

I have shown you all four different techniques that work quite well for various design scenarios. I hope that this has started to get the thought process churning to think of ways that you can add intelligence to your designs and workflows.

Remember these questions to ask yourself to see if you should use any of these workflows.

- Is the work repetitive?
- Is there a need or desire to improve consistency?
- Do you want to improve efficiency?
- Do you need to create many models in a short amount of time?

Remember this; experience with 3D CAD applications is an important factor when it comes to developing good workflows that are geared towards your company's deliverables. Product designs help produce new workflows and better ideas to assist in accuracy and speed to market.

My advice to you all is stay current with best practice tips and tricks by reading blogs, articles on-line, and magazines. Also, be sure to stay current with:

Autodesk Websites / Forums:

Autodesk Community Forums: <https://forums.autodesk.com/>

Autodesk Exchange Apps: <https://apps.exchange.autodesk.com/en>

Inventor Blogs:

From the Trenches with Autodesk Inventor (Curtis Waguespack):

<http://inventortrenches.blogspot.com/>

The CAD Setter Out (Paul Munford):

<https://cadsetterout.com/>

Inventor Tales (Jonathan Landeros):

<http://www.inventortales.com/>

