

MFG321077

## **Making the Content Center Do More for You**

Pete Strycharske  
D3 Technologies

### **Learning Objectives**

- Eliminate cumbersome iPart placement by publishing large iPart tables as Content Center families
- Control component numeric values with custom column expressions
- Utilize external Excel spreadsheets to drive data consistency
- Automatically integrate part numbers in Tube & Pipe conduit data

### **Description**

The Content Center is an invaluable Inventor tool primarily used to store standard components and drive design accelerators, such as Frame Generator, Tube and Pipe, and Bolted Connections. Due to the Content Center database structure, there are several advantages for product design and data manipulation. One advantage is to replace large iPart tables with Content Center families. Simply publishing these large families to a custom Content Center Library will greatly reduce the amount of time required to place and modify these components versus placing them via the iPart table. We'll also explore customization techniques offered in the Family table, but particularly via the Microsoft Excel spreadsheet linking. Editing Family tables within Excel unleashes all the power the spreadsheet tools have to offer, simplifying data coordination. Finally, we'll use iLogic to extend the data within the Family table to solve the vexing problem of automatically establishing part numbers for Tube and Pipe Conduit parts.

### **Speaker(s)**

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. 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! Excited to teach at Autodesk University for the second year in a row and always looking to learn more from all the excellent sessions!

## Table of Contents

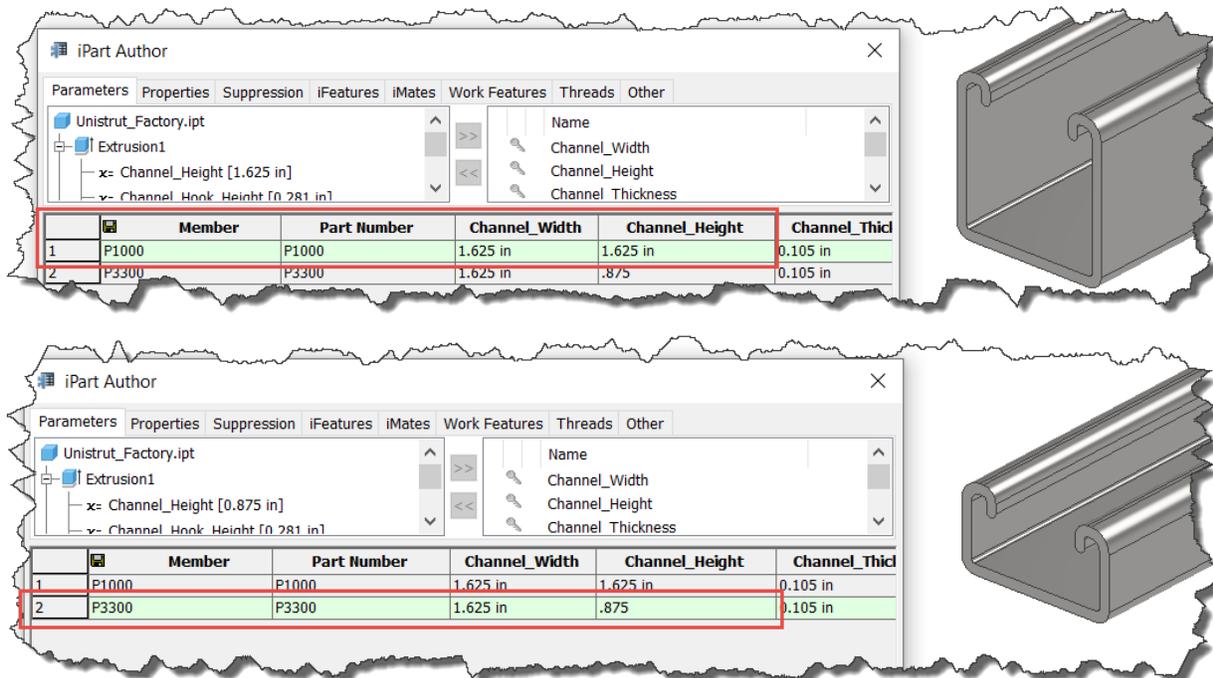
Eliminate cumbersome iPart placement .....	3
What is an iPart and how does it work?.....	3
Summary of iPart information .....	5
What is a Content Center Family and how does it work?.....	5
Summary of Content Center information .....	6
When to use iParts or Content Center Families.....	7
Summary of iPart usage .....	8
Summary of Content Center Family usage.....	8
Special case: Large iPart tables .....	8
Summary of large table performance .....	10
Control component numeric values with custom column expressions .....	10
Publish the controls from an iPart table .....	10
Create and modify the controls within a Content Center Family .....	14
Summary of numeric controls within a Content Center Family.....	15
A word on using iLogic with Content Center components.....	15
Summary of iLogic rules within Content Center Families.....	19
How to utilize external Excel tables to drive data consistency .....	19
Controlling Content Center data via Excel.....	19
Summary of Excel edits to Content Center Family table.....	21
The why and how of storing the external Excel spreadsheet.....	21
Summary of saving and storing Excel spreadsheet for future Family edits .....	22
Accessing the external Excel spreadsheet for Content Center updates.....	22
Summary of using Excel to drive Content Center edits.....	24
How to automatically integrate part number data into Tube & Pipe Conduit Parts Lists.....	25
Current Inventor methodology .....	26
Summary of the Current Method .....	28
Considerations inside of the Content Center Family table .....	28
Summary of Content Center Considerations .....	31
iLogic rule to get the job done .....	32
Summary of iLogic rule.....	33
Setting up and triggering the external iLogic rule.....	33
Summary of configuring external iLogic rules .....	36
Bonus Topic: A workflow for moving a Content Center Family .....	37
The Workflow .....	37
Important Considerations for a Content Center Family move .....	40
Appendix A: Weblinks to technical articles .....	42
Appendix B: Weblinks to video demonstrations.....	43
Appendix C: iLogic rule text.....	44

## Eliminate cumbersome iPart placement by publishing large iPart tables as Content Center families

iPart tables perform a valuable service within Inventor by providing a platform for users to create a “family” of parts, all based on a common design theme. Content Center also provides a similar platform but handles the data differently from the iPart table. We’ll briefly explore both iParts and the Content Center to determine when to use each of these tools

### What is an iPart and how does it work?

I mentioned that an iPart is a table of components that are, typically, closely related geometrically. A common example that may be familiar to many of you is a Unistrut® member, which varies in width, height, thickness, length and can incorporate several different features (holes, slots, etc.). As you can see in the figure below, the values in the iPart table directly impact the component configuration. We’ll be revisiting this design throughout the presentation.



Results of Unique iPart Row Information

An interesting fact, that I recently learned from someone at Autodesk, is that the iPart data is actually stored inside of an embedded Excel spreadsheet. This is part of the reason why Excel is required on computers where Inventor is installed. Utilizing Excel is a great way to quickly and easily modify or add iPart data, with the added benefit of using Excel’s powerful cell formatting capabilities. For some additional information and an example of Excel being utilized to control the iPart table, please refer to [this D3 blog post](#). The screenshots below demonstrate how I’ve utilized Excel to tailor the information for the Unistrut iPart table.

Formula Bar: `=SUBSTITUTE(G3," mm","")`

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Member<default	Part Number [Project]	Channel_Width	Channel_Height	OA_Part_Length	Operation	PN_Length	Length_Number	Profile_Desig				
2	P1000_0150N	P1000_0150N	1.625 in	1.625 in	150 mm	None	0150	150	P1000				
3	P1000_0125N	P1000_0125N	1.625 in	1.625 in	125 mm	None	0125	125	P1000				
4	P1000_0150T	P1000_0150T	1.625 in	1.625 in	150 mm	T_Slot	0150	150	P1000				
5	P3300_0150N	P3300_0150N	1.625 in	.875	150 mm	None	0150	150	P3300				

Use the SUBSTITUTE Formula in Excel to Remove the Units (If Necessary)

Formula Bar: `=TEXT(L3,"0000")`

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Member<default	Part Number [Project]	Channel_Width	Channel_Height	OA_Part_Length	Operation	PN_Length	Length_Number	Profile_Desig				
2	P1000_0150N	P1000_0150N	1.625 in	1.625 in	150 mm	None	0150	150	P1000				
3	P1000_0125N	P1000_0125N	1.625 in	1.625 in	125 mm	None	0125	125	P1000				
4	P1000_0150T	P1000_0150T	1.625 in	1.625 in	150 mm	T_Slot	0150	150	P1000				
5	P3300_0150N	P3300_0150N	1.625 in	.875	150 mm	None	0150	150	P3300				

Use the TEXT Formula in Excel to Format the Part Length Utilized in the Part Number

Formula Bar: `=CONCATENATE(M3,"_",K3,LEFT(J3,1))`

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Member<default	Part Number [Project]	Channel_Width	Channel_Height	OA_Part_Length	Operation	PN_Length	Length_Number	Profile_Desig				
2	P1000_0150N	P1000_0150N	1.625 in	1.625 in	150 mm	None	0150	150	P1000				
3	P1000_0125N	P1000_0125N	1.625 in	1.625 in	125 mm	None	0125	125	P1000				
4	P1000_0150T	P1000_0150T	1.625 in	1.625 in	150 mm	T_Slot	0150	150	P1000				
5	P3300_0150N	P3300_0150N	1.625 in	.875	150 mm	None	0150	150	P3300				

Use the CONCATENATE Formula in Excel to Build More Complex Part Numbers

	Member	Part Number	Channel_Width	Channel_Height	Channel_Th
1	P1000_0150N	P1000_0150N	1.625 in	1.625 in	0.105 in
2	P1000_0125N	P1000_0125N	1.625 in	1.625 in	0.105 in
3	P1000_0150T	P1000_0150T	1.625 in	1.625 in	0.105 in
4	P3300_0150N	P3300_0150N	1.625 in	.875	0.105 in

Options... Verify

Results of the Excel Edits, Viewed Inside of Inventor (Reddish Cell Colors, Indicating Formula Usage)

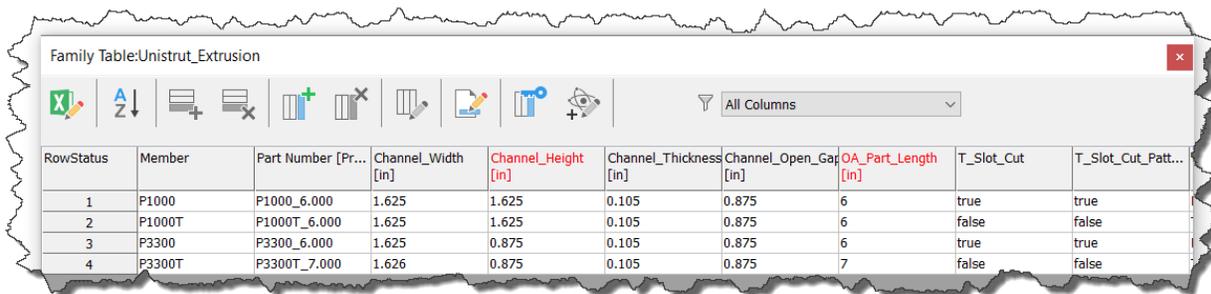
### Summary of iPart information

- iParts are families of similar components that are controlled via a single parent file
- iPart tables are actually Excel tables imbedded inside the Inventor file
- iPart tables can be modified inside of Excel and will retain Excel formulas and formatting

### What is a Content Center Family and how does it work?

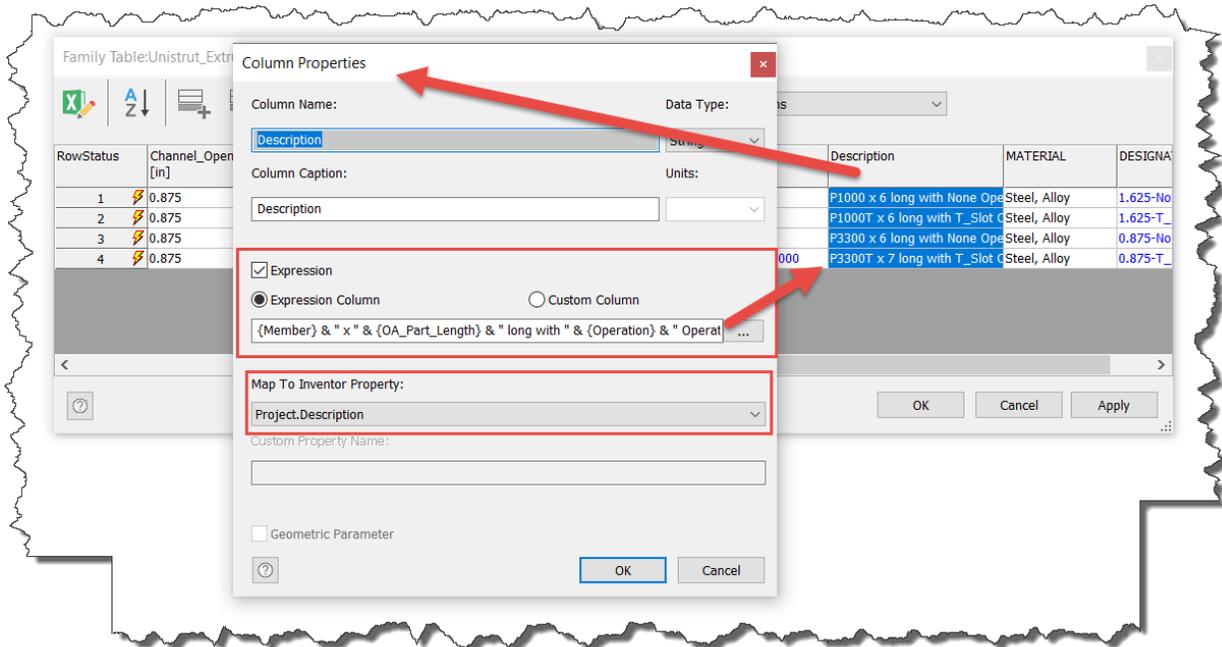
A Content Center Family is also a family of components, but instead of being based on an imbedded Excel table, like an iPart, these families are databases, managed via SQL. This utilizes all the data access advantages that SQL databases, which will come into play when dealing with very large tables.

Content Center Families are accessed via the Content Center Editor, where data can be modified inside the large table. The family table columns can have the data edited cell by cell, or expressions can be applied to every cell simultaneously. This table can also be accessed and modified via Excel, but since the family is not an embedded table, any formatting or formulas that are added in Excel are NOT retained inside the family. Therefore, any subsequent Excel edits to the family table, will not present the formulas. This can be a limitation, which will address a bit later.



RowStatus	Member	Part Number [Pr...	Channel_Width [in]	Channel_Height [in]	Channel_Thickness [in]	Channel_Open_Gap [in]	OA_Part_Length [in]	T_Slot_Cut	T_Slot_Cut_Patt...
1	P1000	P1000_6.000	1.625	1.625	0.105	0.875	6	true	true
2	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	false	false
3	P3300	P3300_6.000	1.625	0.875	0.105	0.875	6	true	true
4	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	false	false

Image of Family Table Editor



Editing all cells at once with the Column Properties Editor

AutoSave  Off    Content Center Family Table for Unistrut\_Extrusion (db441062-d629-4778-bd5f-5dd7aaa2c912).xlsx...

File Home Insert Page Layout Formulas Data Review View Help Acrobat Search

Clipboard Font Alignment Number Styles

RID	C	D	E	F	G	H	I	J
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length	T_Slot_Cu
3	P1000	P1000_6.000	1.625	1.625	0.105	0.875	6	true
4	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	false
5	P3300	P3300_6.000	1.625	0.875	0.105	0.875	6	true
6	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	false
7								

Editing all the Family Table via Excel

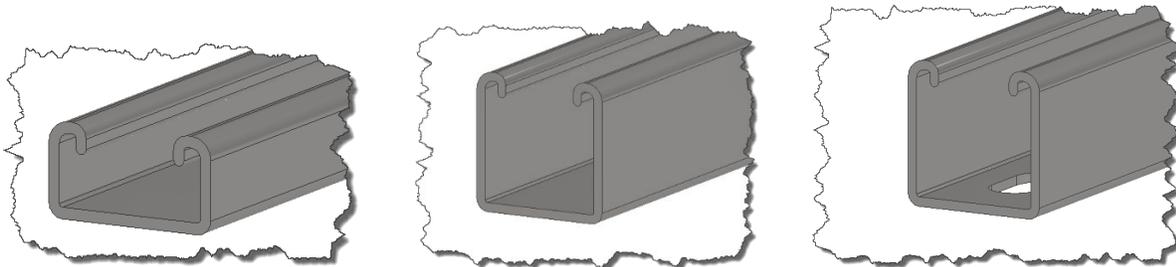
### Summary of Content Center information

- Content Center Families are a family of components that are controlled from a central table
- Families are actually database files managed via SQL
- Primary and expansive edits can be completed inside the Family editor
- Modifications can be made via Excel, but custom Excel formatting and formulas are NOT retained upon completion of the edits

### When to use iParts or Content Center Families?

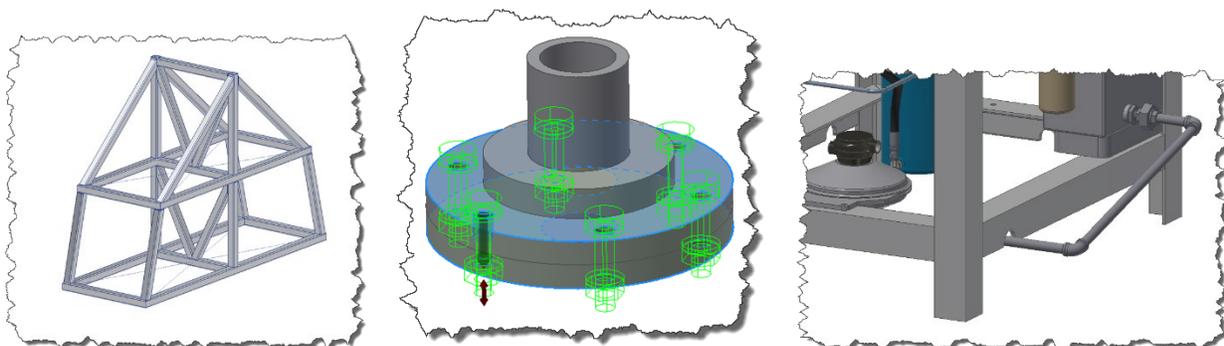
We've covered some of the important characteristics of iParts and Content Center families, but naturally the next question is when should one be used versus the other. In general, while iParts and Content Center families can both be used for most design tasks, each has its advantages in given situations.

iParts are useful because most everyone on a design team can create and utilize them as needed, without lots of authoring or going through the process of publishing to the Content Center, which can be quite useful for custom design work or conceptual design. This can also be a bit easier to visualize and experiment with, as a component family is being developed. If the family of components is constantly growing with additional variations, then an iPart can be a better fit, due to the ease of adding additional information.



Examples of iPart members

Content Center families are easier to centralize and ensure that everyone in the design team is working off a common platform of component designs and can also be easier to manage and update. Some modules in Inventor, such as: Tube & Pipe Environment, Bolted Connections and Frame Generator; must use components from the Content Center. Also, as we'll see in the next section, that large tables are also a better fit within a Content Center family.



Frame Generator, Bolted Connection & Tube and Pipe examples

### Summary of iPart usage

- Easier to create for most team members and more visual in nature
- No publishing required
- Can be easier to create additional row variations

### Summary of Content Center Family usage

- Provides a platform for consistent team design
- Can be easier to manage
- Families required for assembly Design Accelerators

### Special case: Large iPart tables

There is one large advantage that Content Center families have over iPart tables; large table sizes run much more quickly when Content Center families as opposed to iPart tables. I found this result initially a bit surprising, but after performing some research, which we'll cover shortly, this makes much more sense.

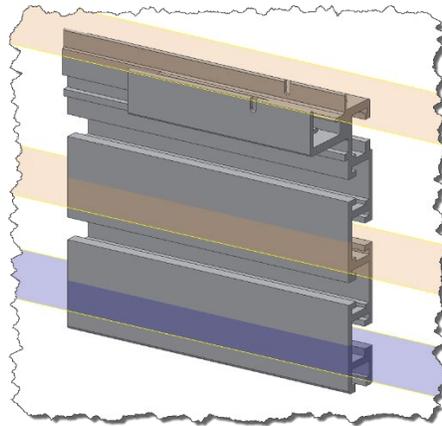
However, before we get into that discussion, I want to take a moment to thank thank the design team at Shuttleworth, a brand within the ProMach company, who is letting me use some of their data for this presentation. They are one of my customers and I'm super grateful that they brought this issue of large iPart tables to my attention, so that I could help them become more efficient and that they were willing to aid me in this AU class!

The problem that Shuttleworth brought to my attention was twofold. First of all, one of the iPart files they shared with me had a table topping out at **4,416 ROWS!!!** This is a massive iPart table, probably the largest one I've ever personally worked with. Secondly, their part lengths, which are in mm, are always shown in the part number with 4 digits. For example, if the length is 65 mm, then the part number will show a "-0065" for the length value. This second requirement will be addressed in more detail in the 2<sup>nd</sup> and 3<sup>rd</sup> sections of this report.

Unfortunately, due to the size of the iPart table, the part was becoming pretty unbearable to work with. Here are a list of some metrics that I gleaned from my work on the part:

iPart Table Operations & Timing	
Time (min:sec)	Operation
0:51	To edit the iPart table and make a simple change
6:04	To open up the Excel table to edit the iPart and return to Inventor
3:31	Configure and place the iPart into an assembly file

Now these times can be improved with high powered hardware and if you've been loading up the Excel time at least once before, but my times are pretty representative of just firing up Inventor and getting started right away in the morning. Performance like what is shown above can put one in a "fun" mood for the rest of day or provide ample time to pick up coffee and donuts. In short, these times are not great...



Sample Shuttleworth iPart with 4,416 rows

The Shuttleworth team had come to the conclusion that work could not continue in this fashion, with this iPart and so they turned to us at D3. I had done a fair amount of work in the past with the Content Center, so I acquired the sample and published it to the Other category within the Content Center. This allowed us to test out the concept without having to reconfigure any data to satisfy the requirements for the Structural Content Center category. The results were quite excellent, as shown in the table below.

Content Center Family Operations & Timing	
Time (min:sec)	Operation
2:44	Time to publish the iPart to the Content Center
1:04	To edit the Family table and make a simple change
5:53	To open up the Excel table to edit the Family and return to Content Center
0:32	Configure and place the Family member from Content Center into an assembly file

As we can see in the table, loading up and editing the tables takes quite a bit of time (forever, really in the CAD world...) and the times between iPart and Content Center are pretty comparable, although there is the extra time of initially publishing the iPart into the Content Center. However, the really big win is the time it takes to place the components into the assembly, the really day-to-day operation. When placing out of the Content Center I was able to shave off almost a **FULL 3 MINUTES** as compared to placing as an iPart.

My conclusion is to switch to a Content Center family approach if your iPart placement time starts to reach the frustration level. The Content Center Families, due to their being an actual database, can parse through and filter results much more quickly than the iPart tables, which are embedded Excel tables. For a helpful article on the differences between Excel and database tables, please see this helpful [Codecademy article](#).

### Summary of large table performance

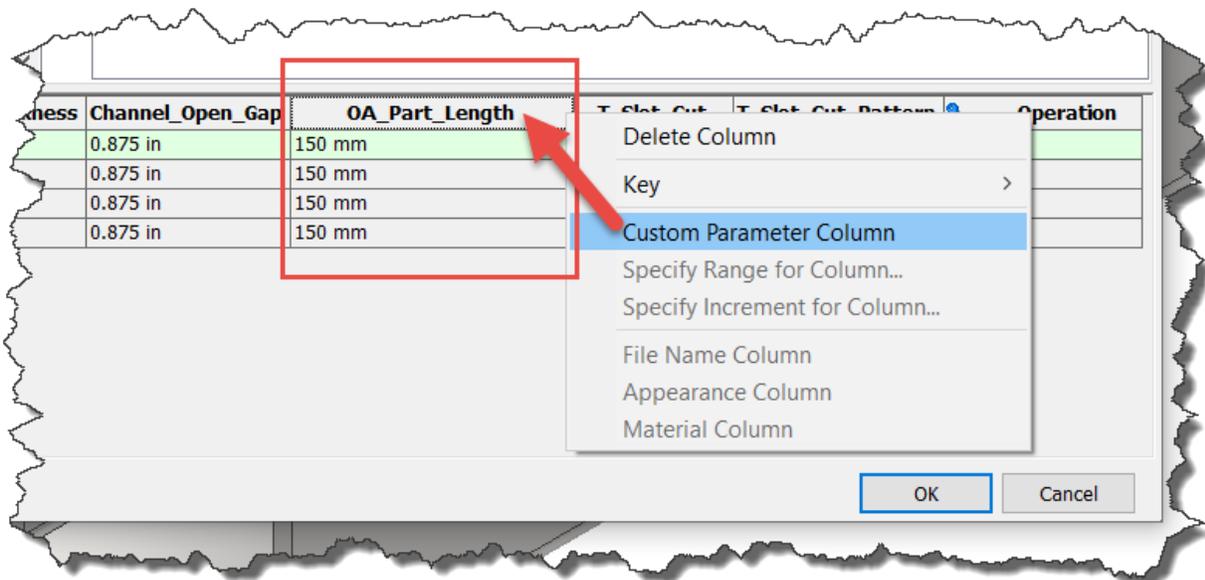
- Editing times between iParts and Content Center families are comparable
- Placing Content Center Families is SIGNIFICANTLY faster compared to iParts

## Control component numeric values with custom column expressions

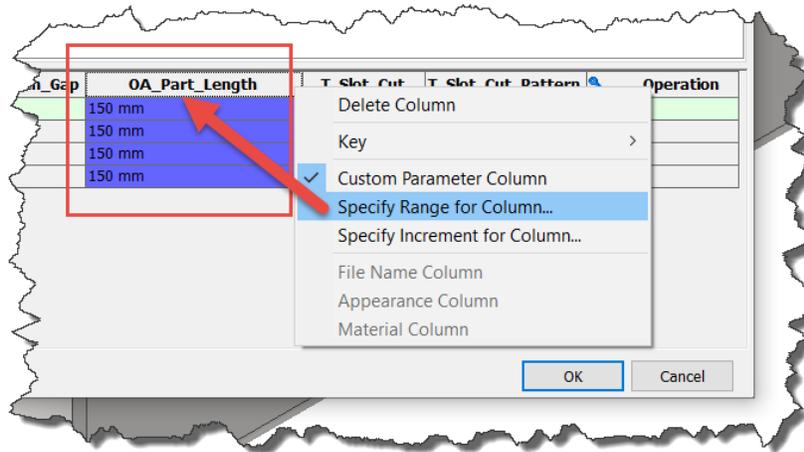
In the first portion of this class, we discussed the transition from iPart table to Content Center Family when dealing with larger amounts of data, but what if the data didn't have to become quite so large? Often the reason that tables become so large is every component size, length typically, is added to the table, necessitating another row. While there can be good reasons for this, which we'll explore in the 3<sup>rd</sup> portion of our class, if this tendency can be avoided, the process for building and maintaining the tables can be greatly simplified and time requirements reduced. Since we've already seen the advantages of utilizing the Content Center Family for large data tables, we'll be focusing in Families in this section.

### Publish the controls from an iPart table

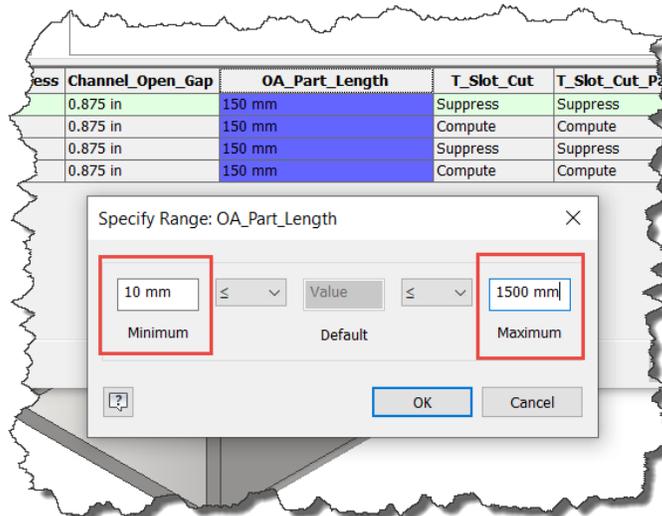
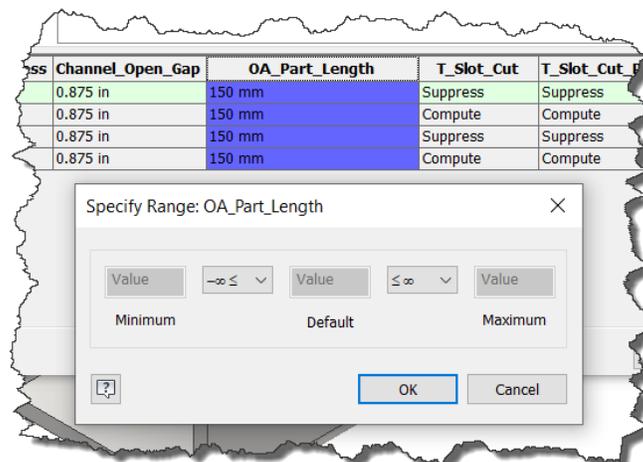
iPart tables have built in controls that can be utilized to help control the numeric values and, thankfully, they're super easy to apply. The screenshots below illustrate the process well and these controls can and should be used for custom iParts.



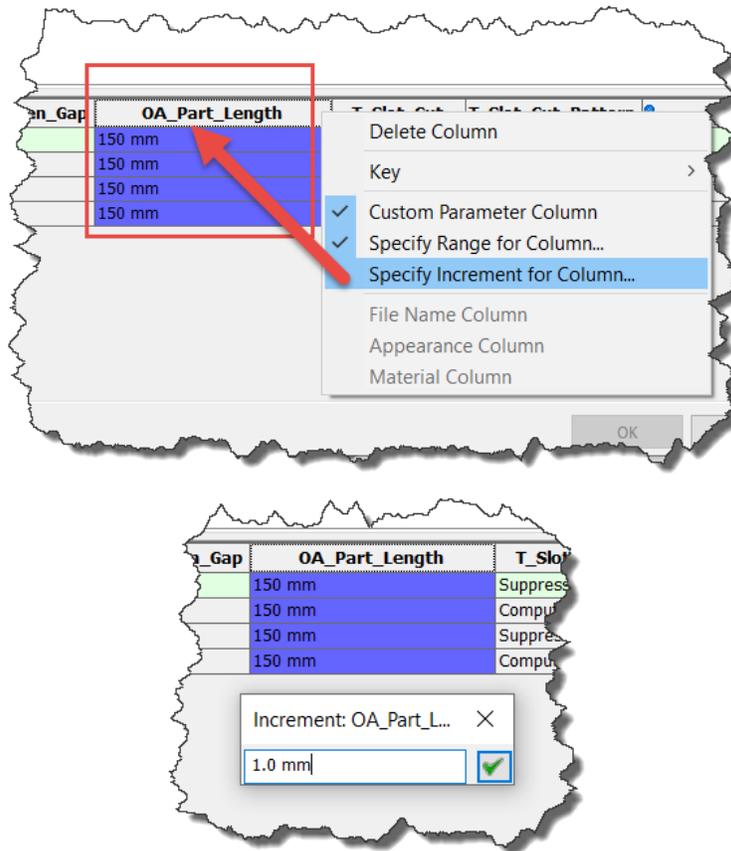
Edit the iPart table, ensure column is NOT a key column, right click and set it to be a "Custom Parameter Column"



Right click to set a “Specific Range” for the custom column



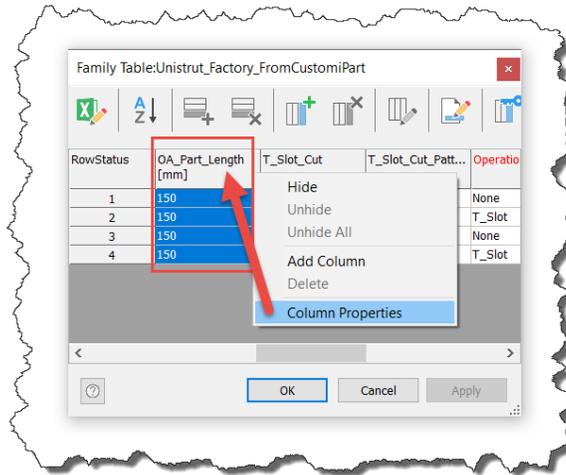
Change the initially unlimited range into defined minimum and / or maximum values



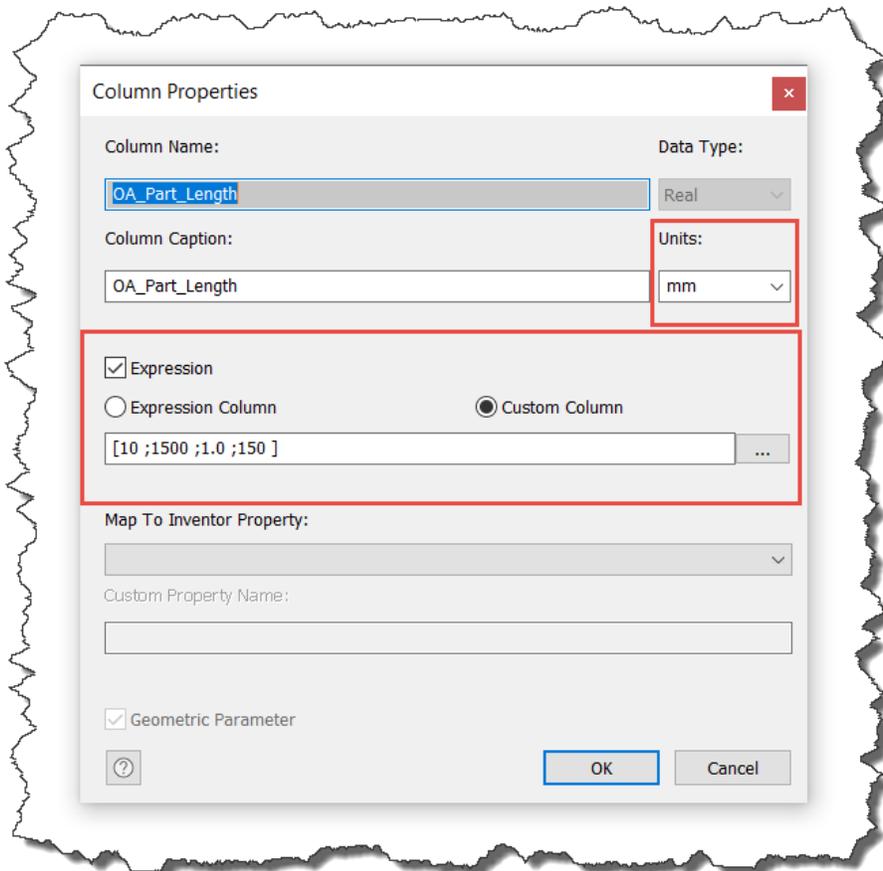
Right click to “Specify Increment Value” for the custom column

Once the custom iPart column is configured as desired, the part can be published to the “Other” category inside the Content Center. The custom column value range and incremental value are populated into the Content Center Family’s table, as a Custom Column Expression. We’ll explore the format and meaning of this custom column expression when we create the controls inside the Content Center Family from scratch.

Please note that when publishing custom iParts to the Content Center, the Member names and Part Numbers will almost certainly NOT be configured correctly. Therefore, one will have to edit the Family table to ensure that duplicate file name madness doesn’t ensue.



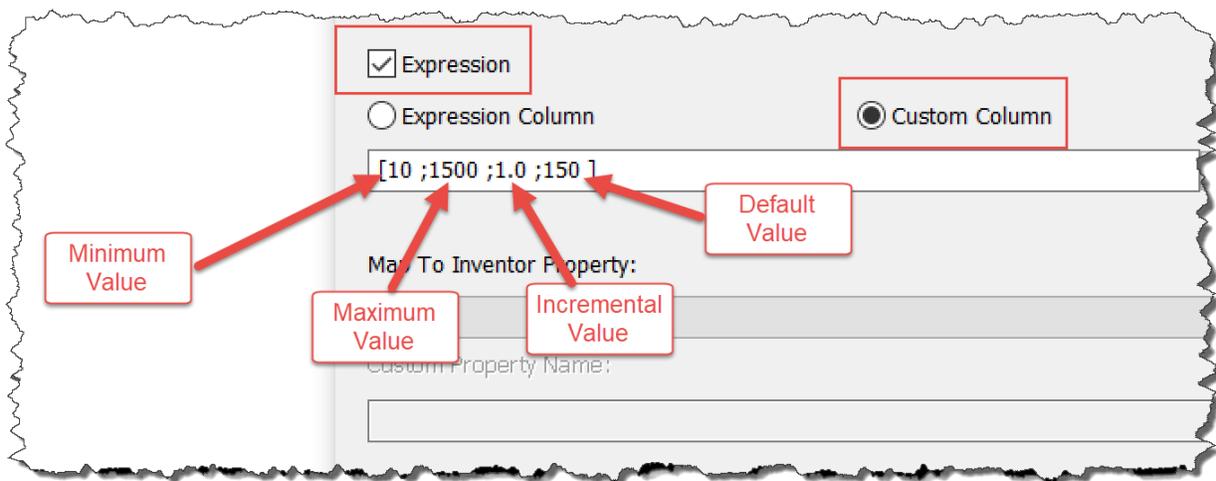
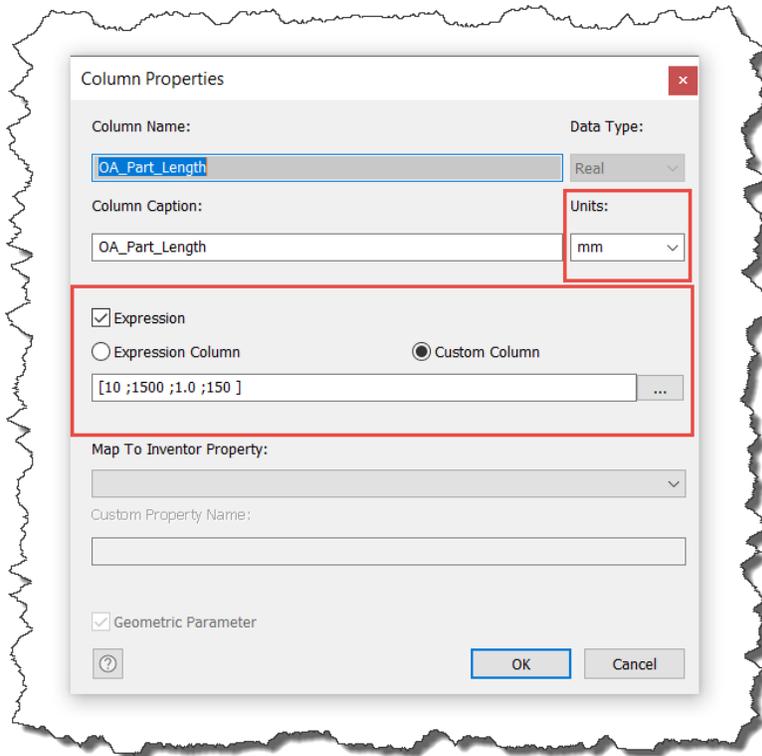
Edit the Content Center Family, right click and select “Column Properties” for the desired column



Ensure the units are correct and notice the expression is prepopulated from the iPart publishing process

### Create and modify the controls within a Content Center family

The Custom Column Expression can also be generated directly within Content Center Family without the use of a custom iPart. The process for editing the Family table remains the same, no matter how the Family is published, but a brief explanation of the values shown in the Custom Column Expression will prove helpful.



Use the above figure to understand how to control user input values during assembly placement

No matter which method is utilized, the user will only be able to enter in approved values based on the Custom Column Expression, so problematic values or sizes can be avoided. However, there is one potential drawback using this method, the component values are whatever the user keys in during placement, so specific character formatting is difficult to enforce. For example, if the part number is supposed to be some combination of a configuration designation and the length, but the length is required to have 4 characters (leading zeros, if necessary), there is no way to force the users to key in something like “0150” for the length in mm. There are a couple of ways to enforce data standards, one with iLogic and in the next section, we’ll revisit the use of Excel to enforce specific value formatting.

### Summary of the numeric controls within a Content Center Family

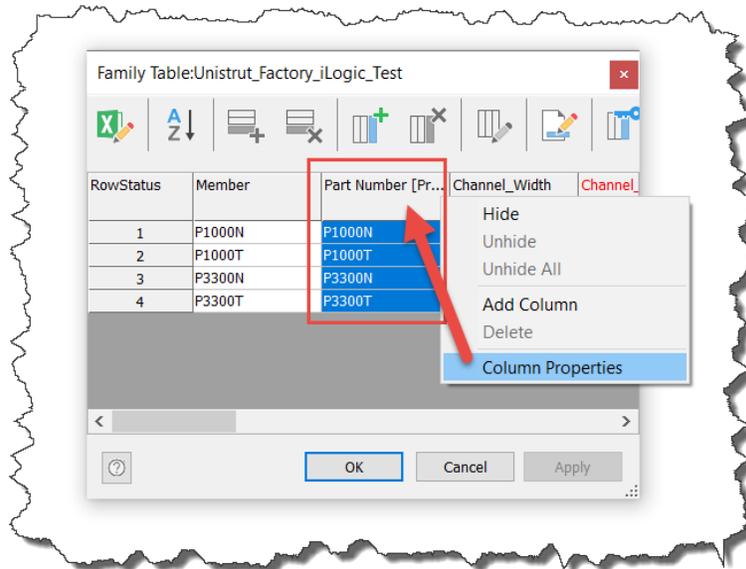
- Numeric controls can be applied when publishing a Family from a custom iPart, where the custom column properties are utilized within the new Family.
- Numeric controls can be applied directly while editing a Family table. The format is {“Minimum Value”; “Maximum Value”; “Increment Value”; “Default Value”} .

### A word on using iLogic with Content Center components

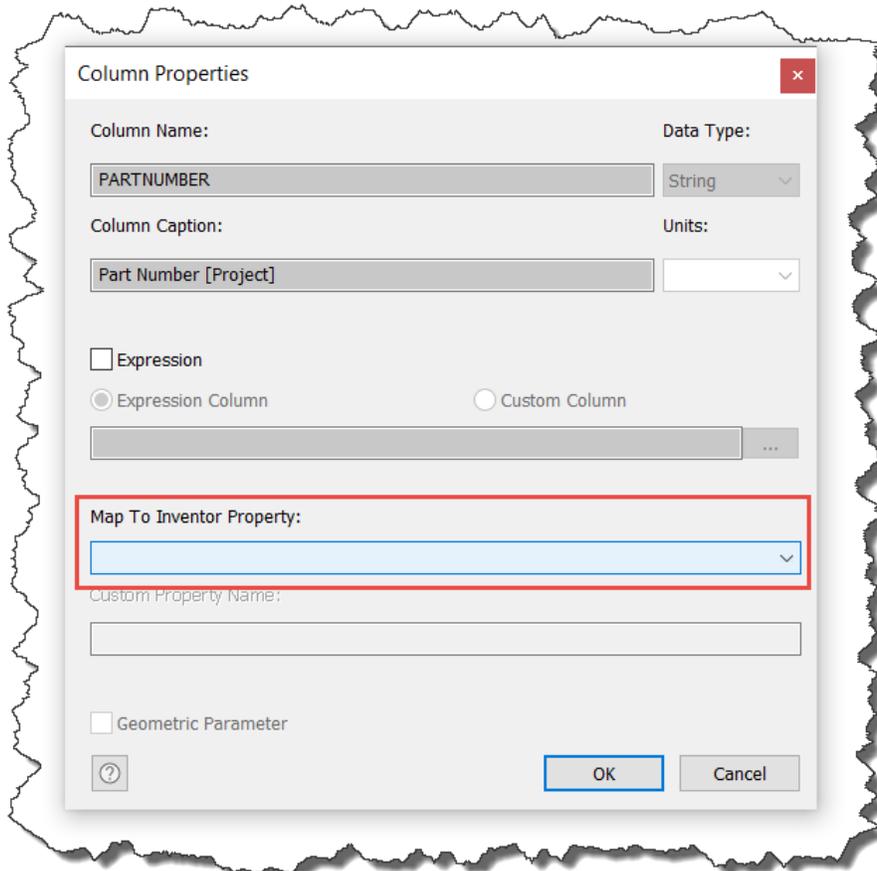
iLogic is a power tool that enables “rules – based” design inside of Inventor. If you’re not familiar with iLogic, think of it as a logical approach to controlling designs; things like “If / Then” logic. iLogic is the one of the primary drivers for design customization and automation. An iLogic rule could be developed to convert the given length of a component, say “150 mm” to a desired character string length, “0150” for example.

iLogic can be utilized within Content Center Families, but one must be **VERY** cautious. If iLogic is to be utilized to control a value within the Content Center member, then the Family table **MUST NOT** also attempt to control that value. Also, one must be careful to trigger the rule at the appropriate time, I typically use the “Before Save” option.

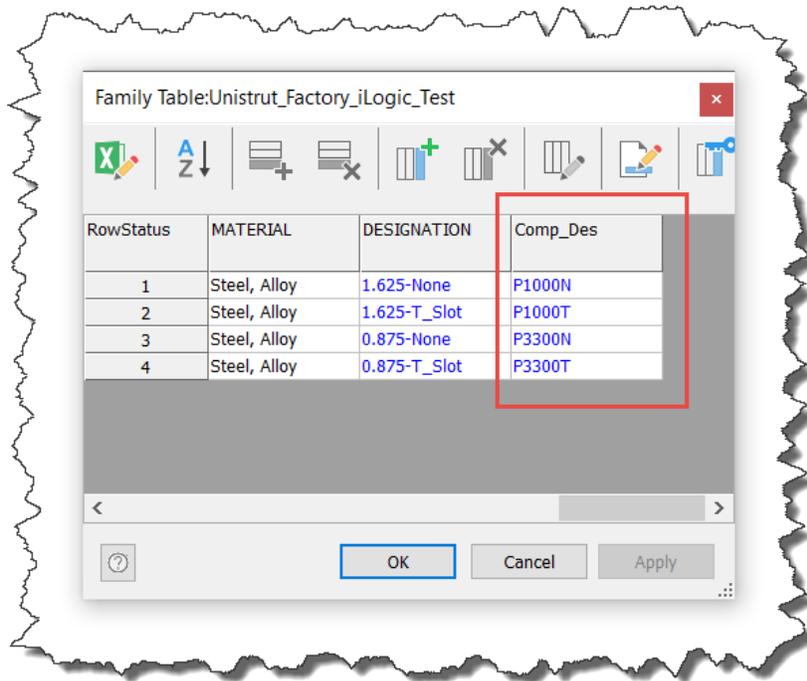
For example, in a Unistrut Family table, I wish to control the part number to include the designation of my row “P1000N”, the P1000 extrusion with no slots. Also, I’d like to include the length of the component, as a four - character string; i.e. “0150”. To accomplish this, I’ll have to build an iLogic rule, which I’ll list in Appendix C, and modify the Content Center Family table’s control over the Part Number. Please see the following figures.



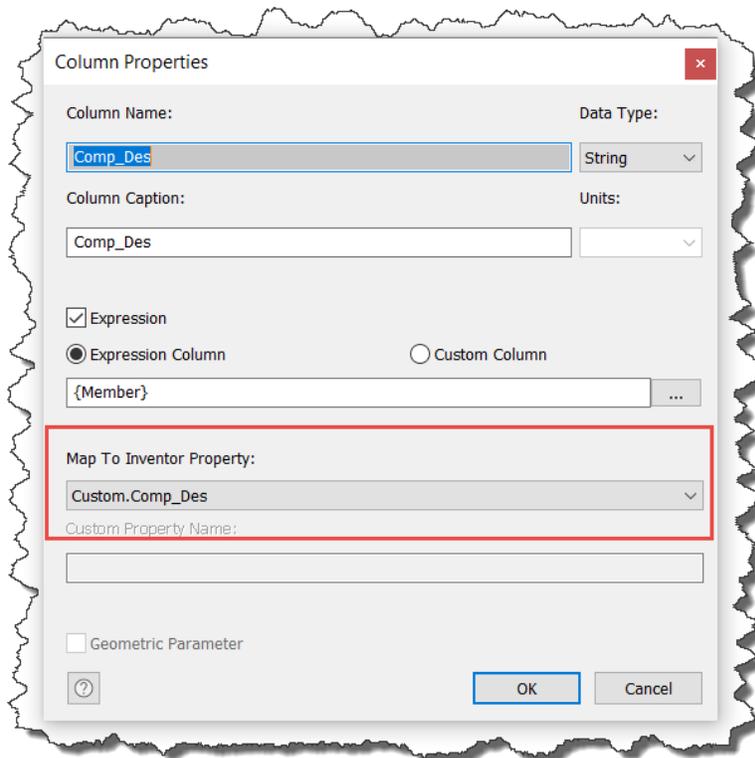
Edit the Family table, right click on the Part Number column heading and select “Column Properties”



Set the “Map to Inventor Property” dropdown to the blank entry, the one at the very top of the list

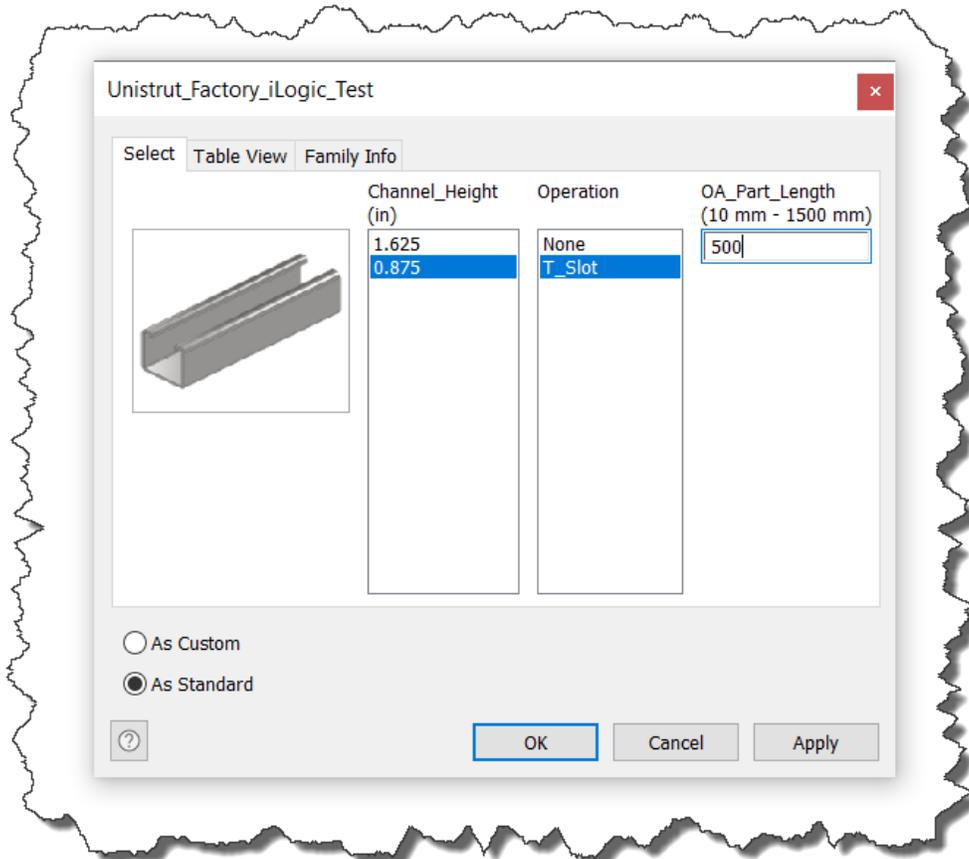


A custom column has been created to track the primary component designation

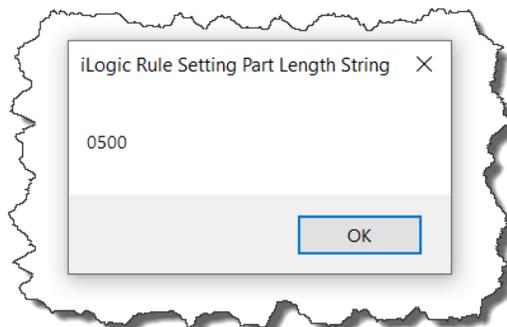


The component designation column is mapped to the corresponding iProperty

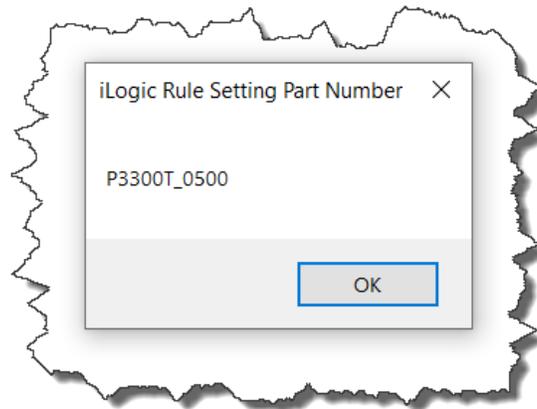
Now that the Family table has been modified, we can utilize a combination of custom iProperties, which can be populated and mapped to directly from the Family table (the reverse of what we just did for the Part Number), and iLogic to create the desired part number.



Configure and place a member from the Content Center



An iLogic rule sets the character string for part length



An iLogic rule combines the component designation iProperty with the formatted part length into a new Part Number

This is a great way to combine a pair of powerful tools: Content Center and iLogic. If one is careful to avoid conflicts between the two, lots of efficiency can be realized.

#### Summary of iLogic rules within Content Center Families

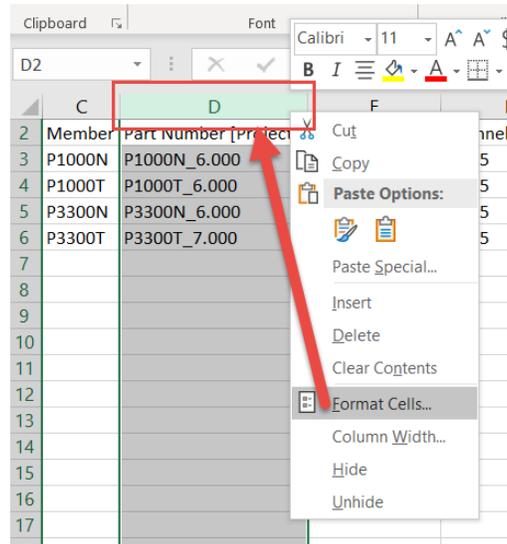
- iLogic is a “rules – based”, logical design engine primarily used to control customizable and automated designs.
- iLogic can be utilized within Content Center Families.
- In order for proper functionality, iLogic rules and Family tables CANNOT both simultaneously be used to control unique component values.

### How to utilize external Excel tables to drive data consistency

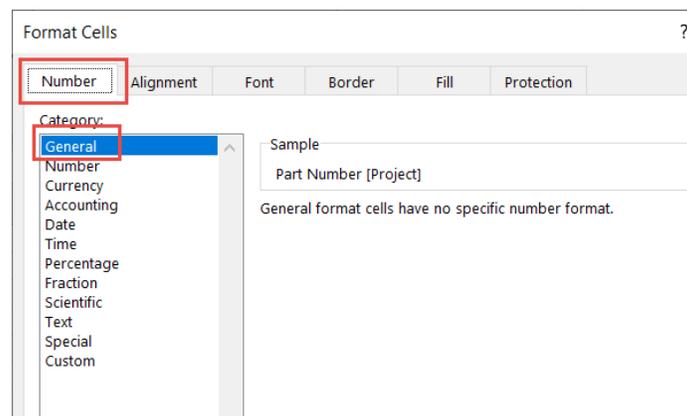
In the last section we learned how to control the values of numeric values using Custom Expressions and examined a scenario where the Part Number iProperty required a specific formatting of the component length. iLogic can fit the bill in some of these situations but won't work all the time. Sometimes the values that we want to specifically format are controlled directly by the table, in which case iLogic cannot really help us, particularly for “Standard” Content Center Families. In these situations, we'll revisit some of the Excel techniques that we saw in the first section of this course.

#### Controlling Content Center family data via Excel

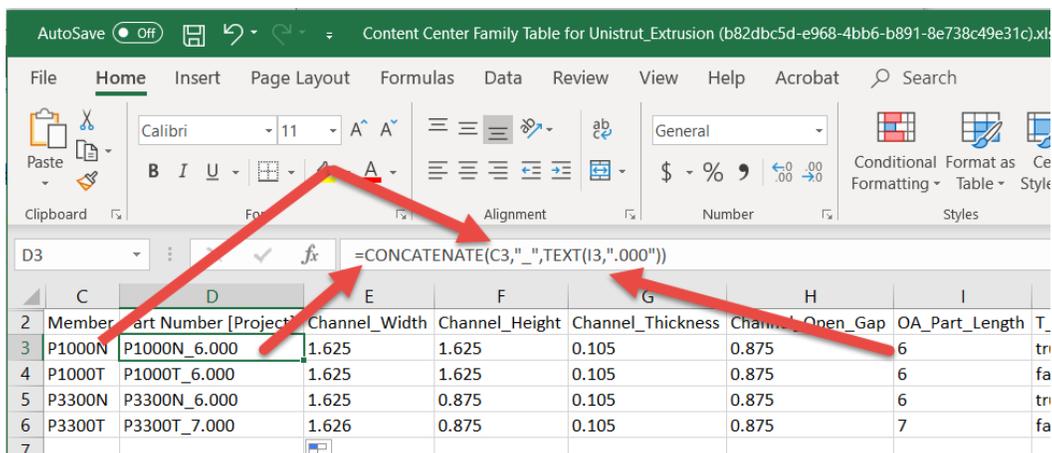
As we saw in the first section of this course, we can quickly and easily edit the table data within a Content Center Family. Excel offers lots of powerful formulas and formatting options to tailor data directly to one's liking. Since we've already look at that in the first section, we won't reexamine those Excel editing steps here in great detail. Unfortunately, however, the custom formatting utilized by Excel to modify the Family table is lost as soon as the Excel sheet is closed. This is due to the fact that the Family table is actually a database table and NOT an imbedded Excel spreadsheet.



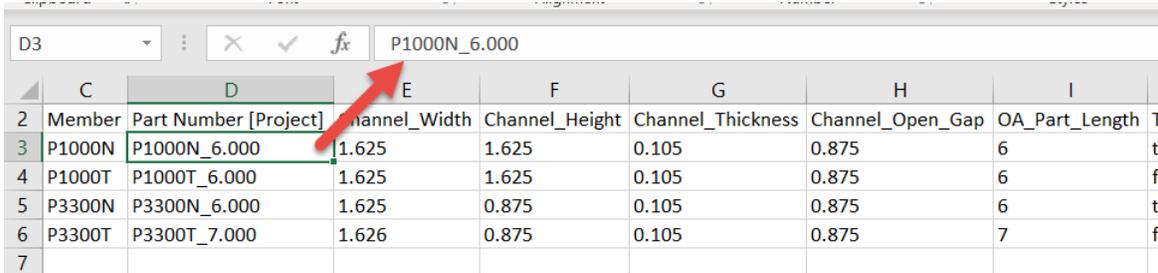
After editing the Family table in Excel, choose a cell in which a formula is to be utilized, right click and choose "Format Cells"



Set the "Number" tab to utilize the "General" format



Utilize the desired formula and data cells, save the spreadsheet and close Excel



	C	D	E	F	G	H	I
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length
3	P1000N	P1000N_6.000	1.625	1.625	0.105	0.875	6
4	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6
5	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6
6	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7

Unfortunately, due to the Family table being a database, the Excel formatting is removed with future edits to the Family table via Excel

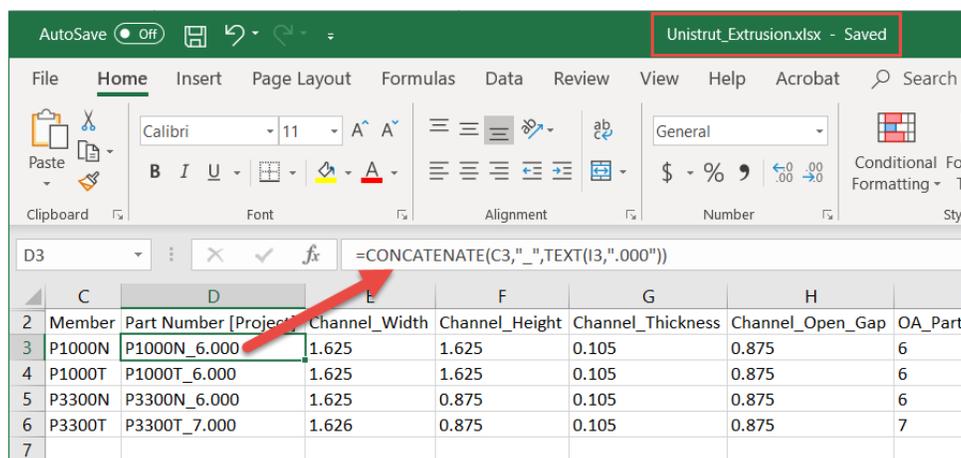
### Summary of Excel edits to Content Center Family table

- Families are actually database files managed via SQL
- Modifications can be made via Excel, but custom Excel formatting and formulas are NOT retained upon completion of the edits

### The why and how of storing the external Excel spreadsheet

Excel has powerful capabilities for controlling the formulation and formatting of data, which can be very useful for complex Family tables. However, the fact that Excel formatting is lost upon completion of the edit and returning to the Content Center severely hampers Excel's effectiveness in this area. Thankfully there is a workflow that will allow us to use and retain the power of Excel for editing the Family tables.

The process for editing the Family table via Excel is also the starting point for this Excel preservation workflow. Immediately after entering Excel to perform the edits, perform a Save As and store the spreadsheet in a network location where all the Content Center admins can access them. Also be sure to name this Excel spreadsheet exactly the same as the Content Center family, which will eliminate confusion later. Finally, apply any desired formulas and / or custom formatting and then save and close the spreadsheet. Moving forward this spreadsheet will be used for edits to the Family table.



	C	D	E	F	G	H	I
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_
3	P1000N	=CONCATENATE(C3,"_",TEXT(I3,".000"))	1.625	1.625	0.105	0.875	6
4	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6
5	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6
6	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7

Save the initial spreadsheet to a standard location and customize as desired

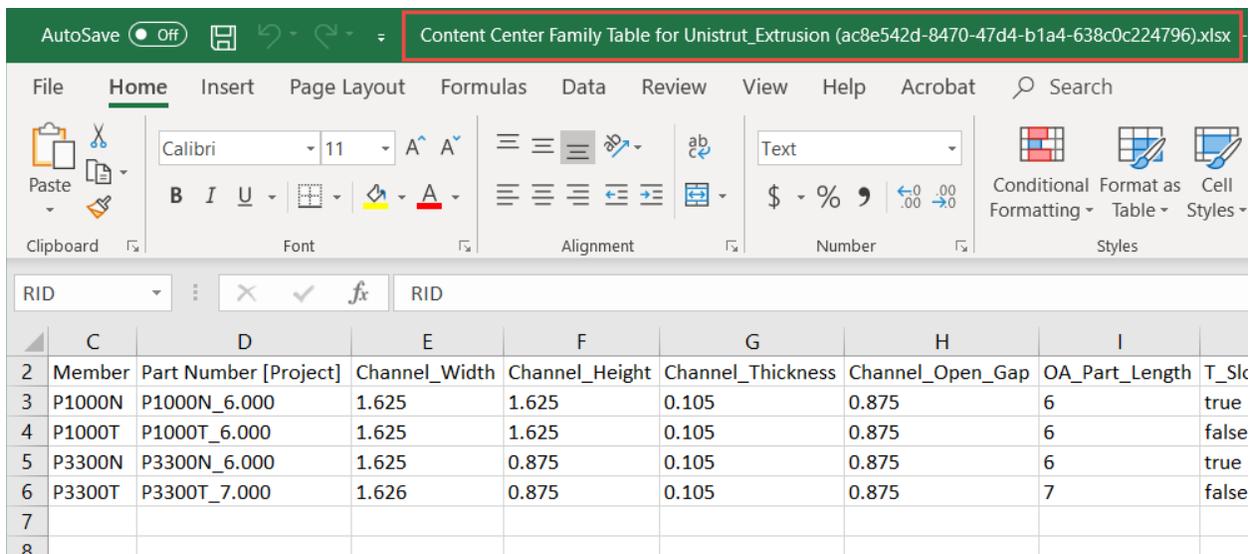
### Summary of saving and storing Excel spreadsheet for future Family edits

- Use the Excel edit option to modify the Content Center Family
- Immediately upon Excel's launch, perform a Save As to store the spreadsheet in a standard location that all Content Center administrators can access
- Be sure the Family and spreadsheet names are exactly the same
- Apply formulas and formatting as desired to the spreadsheet and then close it

### Accessing the external Excel spreadsheet for Content Center updates

Once the spreadsheet has been formatted and saved, it can be used for any future edits. In fact, the spreadsheet now becomes the driving file for all future Family table updates, since the spreadsheet contains all of the critical data formats.

The process for updating the Family table with this external spreadsheet is to kick off an Excel edit of the Family table just like normal. As soon as Excel launches, you must open up the stored spreadsheet from the SAME instance of Excel. Inside the stored spreadsheet, make whatever changes are desired, then select and copy all the data cells (I advise using the CTRL-A and CTRL-C method). Toggle back to the spreadsheet launched by the Family edit, select all the data cells and paste the formatted data cells from the stored spreadsheet (I advise using the CTRL-A and CTRL-V method). Save and close both spreadsheets to finalize and apply the updates to the Family table.



	C	D	E	F	G	H	I	
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length	T_Slc
3	P1000N	P1000N_6.000	1.625	1.625	0.105	0.875	6	true
4	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	false
5	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6	true
6	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	false
7								
8								

Launch the Excel spreadsheet to edit the Content Center Family

AutoSave  Off Unistrut\_Extrusion.xlsx - Excel

File Home Insert Page Layout Formulas Data Review View Help Acrobat Search

Clipboard Font Alignment Number Styles

P6 1.625-T\_Slot-6

	C	D	E	F	G	H	I	T
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length	
3	P1000N	P1000N_6.000	1.625	1.625	0.105	0.875	6	t
4	P1000N	P1000N_10.000	1.625	1.625	0.105	0.875	10	t
5	P1000N	P1000N_18.000	1.625	1.625	0.105	0.875	18	t
6	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	f
7	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6	t
8	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	f

Open the stored spreadsheet, make the desired modifications, select all the data cells and copy them

AutoSave  Off Content Center Family Table for Unistrut\_Extrusion (ac8e542d-8470-47d4-b1a4-638c0c224796).xlsx - ...

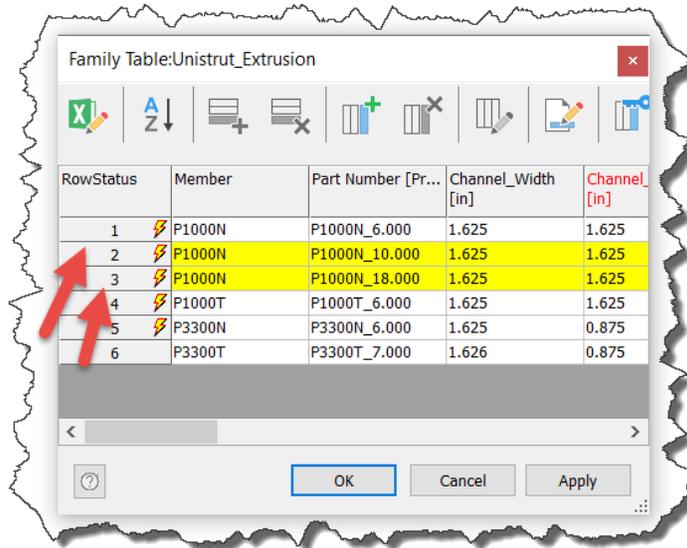
File Home Insert Page Layout Formulas Data Review View Help Acrobat Search

Clipboard Font Alignment Number Styles

A1 RID

	C	D	E	F	G	H	I	J
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length	T_Slot
3	P1000N	P1000N_6.000	1.625	1.625	0.105	0.875	6	true
4	P1000N	P1000N_10.000	1.625	1.625	0.105	0.875	10	true
5	P1000N	P1000N_18.000	1.625	1.625	0.105	0.875	18	true
6	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	false
7	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6	true
8	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	false

Toggle back to the spreadsheet launched from the Family table edit, select all the existing cell data and paste the stored cell data to apply the edits



RowStatus	Member	Part Number [Pr...	Channel_Width [in]	Channel [in]
1	P1000N	P1000N_6.000	1.625	1.625
2	P1000N	P1000N_10.000	1.625	1.625
3	P1000N	P1000N_18.000	1.625	1.625
4	P1000T	P1000T_6.000	1.625	1.625
5	P3300N	P3300N_6.000	1.625	0.875
6	P3300T	P3300T_7.000	1.626	0.875

Save and close both spreadsheets and the updates are applied to the Family table

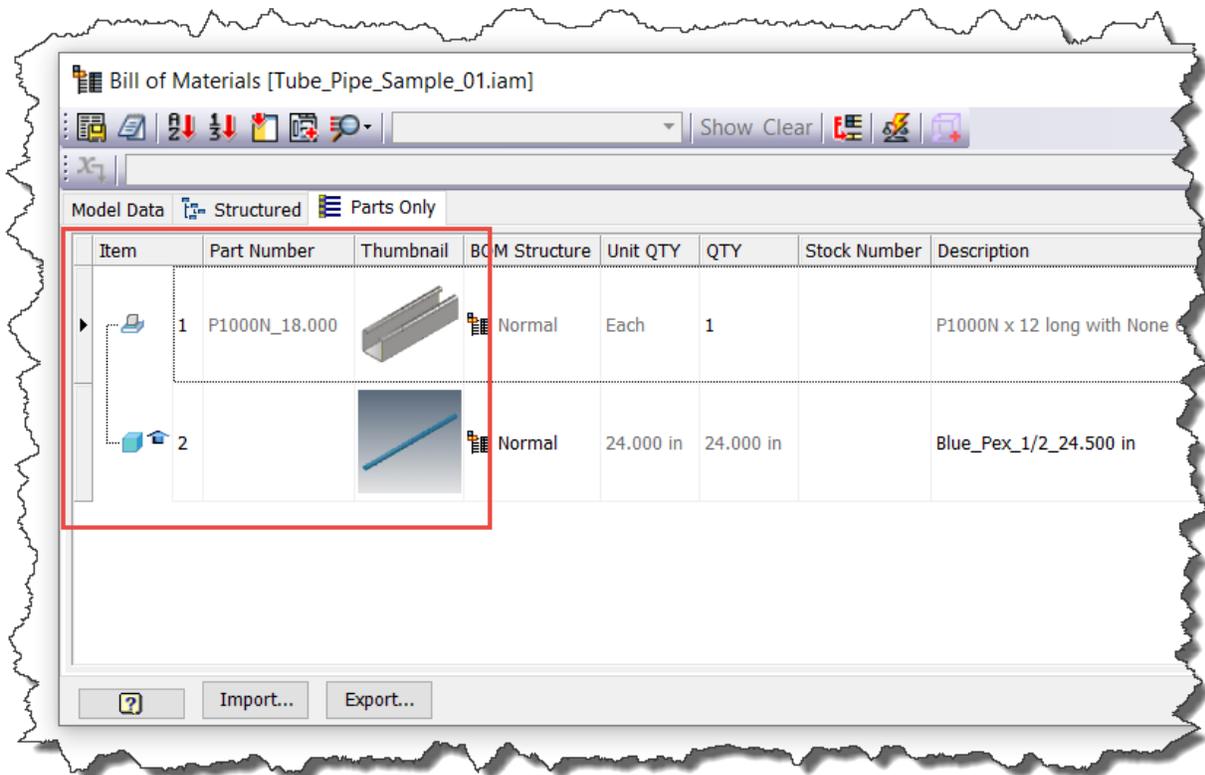
This is a reliable way to make updates to Family tables yet retain the Excel formatting that is desired. In order to avoid confusion, I recommend that admins drive cell data either with Excel formulas and formatting or with Content Center Expressions. While driving some columns with Excel and other columns purely with the Content Center is possible, this approach will take a higher degree of coordination to ensure that data integrity is maintained.

### Summary of using Excel to drive Content Center edits

- Launch the Excel spreadsheet to edit the Content Center as usual
- In the SAME Excel session, load the stored spreadsheet
- Edit the stored spreadsheet as desired
- Select and copy all the data in the stored spreadsheet
- Toggle back to the Family edit spreadsheet
- Select all the initial data and paste the stored data to overwrite the initial data
- Save and close both spreadsheets to apply the updates to the Family table

## How to automatically integrate part number data into Tube and Pipe Conduit Parts Lists

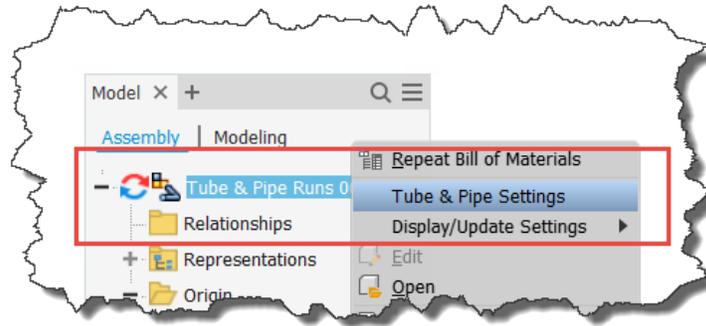
In this last section of the class, we'll break away from our focus on the Content Center Family versus Excel editing and will hone in on a unique situation involving the Tube & Pipe environment. For most components that are placed from the Content Center, the iProperty data is populated automatically, particularly the Part Number. However, as you'll see below, Part Numbers for Piping components are NOT automatically placed when used inside of Tube & Pipe runs. However, by the end of this section, we'll have a workaround that can be implemented with relative ease.



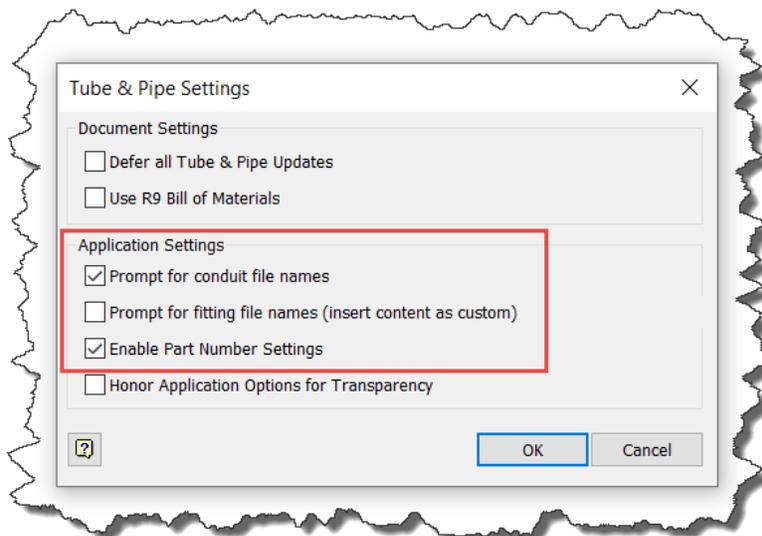
Most components placed from the Content Center will show the Part Number, Tube & Pipe conduit parts do not

### Current Inventor Methodology

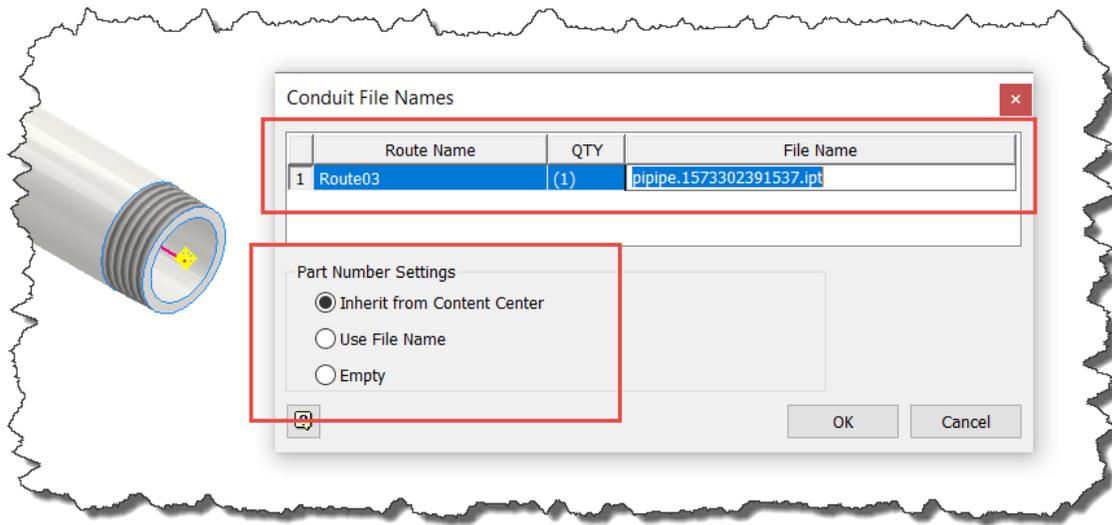
There is a method for applying Part Numbers in out-of-the-box Inventor via the Tube & Pipe Settings within the master runs subassembly. By right clicking on the master runs subassembly and selecting the Settings, one can turn on the ability to name the tube & pipe conduit parts and apply a Part Number as well. Once the routes are populated then the naming conventions become active and can be applied.



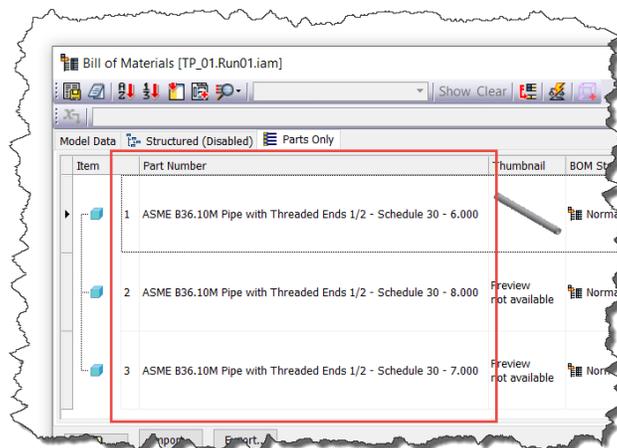
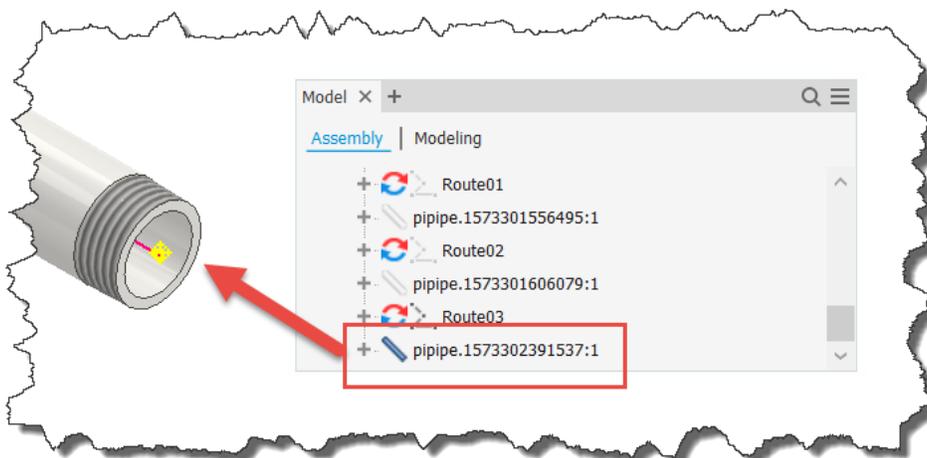
Access the Tube & Pipe Settings



Configure the Conduit Naming and Part Number Settings as Desired



Populating the route allows Conduit Naming and Part Number application



The resulting Conduit Naming and Part Numbers

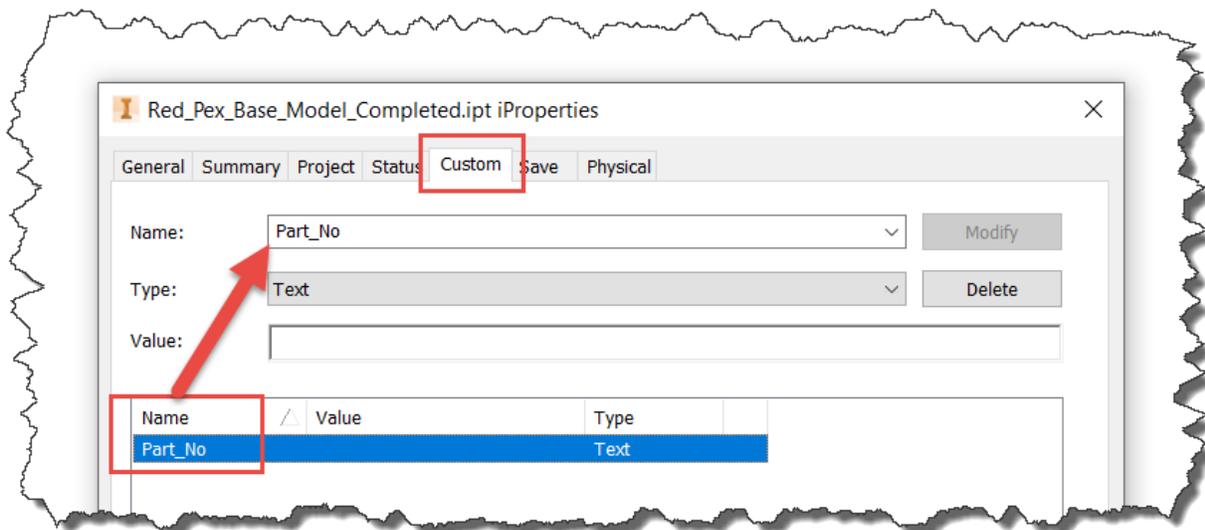
This method does work, but there are a couple of concerns. Firstly, this will allow users the ability to name the individual conduit files, which I'd prefer to avoid, even though the file names still default to the auto-generated values. Secondly, this method only works with newly populated routes, meaning it won't work with routes that already populated members. If a route has already populated members, a new route will have to be generated in order to get this method to work for previous routes. Finally, the assembly browser names become the file name, displaying the auto-generated values, which I don't find very descriptive. Therefore, I endeavored to find this alternate method that we'll discuss in the next few sections.

### Summary of the Current Method

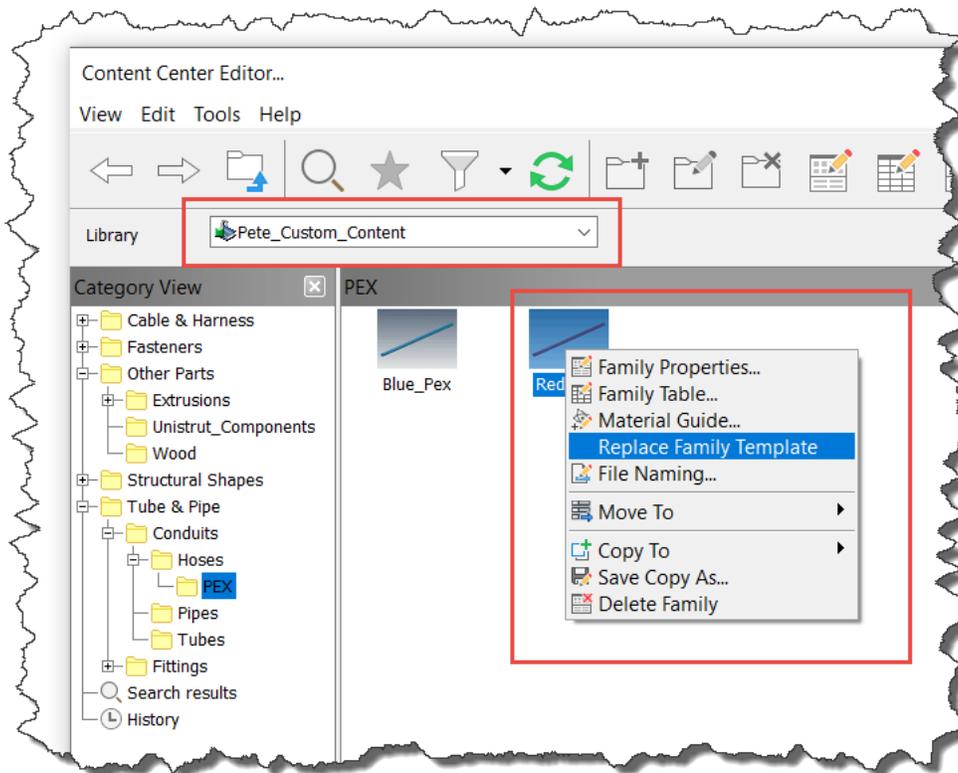
- Configure the Tube & Pipe Settings
- In order to apply Part Numbers, the option to customize File Names must be active
- If existing populated routes are present, a new route must be generated and populated to apply the settings to the existing routes
- Auto-generated File Names become the assembly browser names

### Considerations inside of the Content Center Family table

In order to eventually populate the Part Number iProperty, we'll have to set up a custom iProperty and define its column properties within the Content Center Family table. From there one can publish the conduit part to the Content Center or if the Family already exists, one can replace the family template.



Define the "Part\_No" iProperty, which will eventually carry the Part Number information

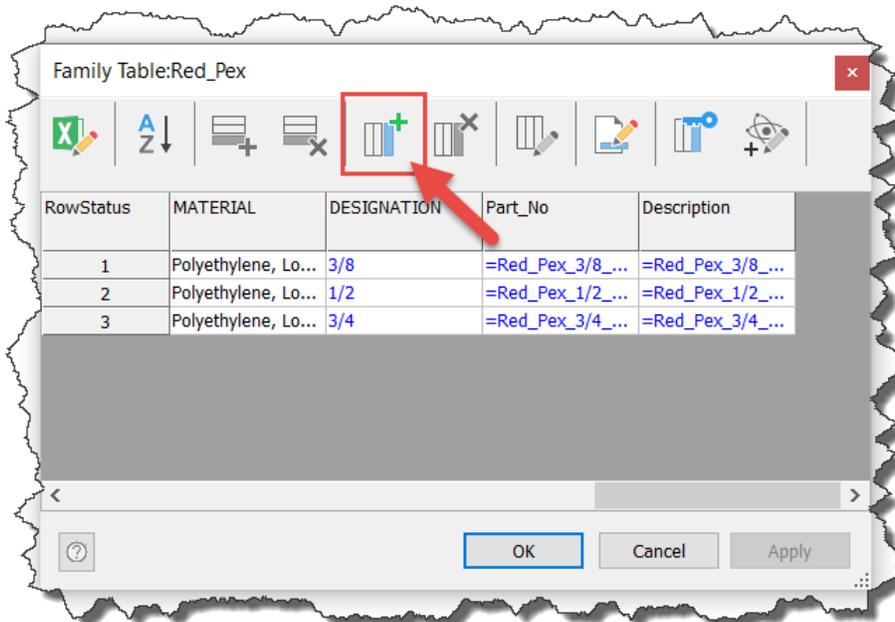


Publish the component to the Content Center of “Replace the Family Template” if the Family table already exists to bring the custom iProperty into the table

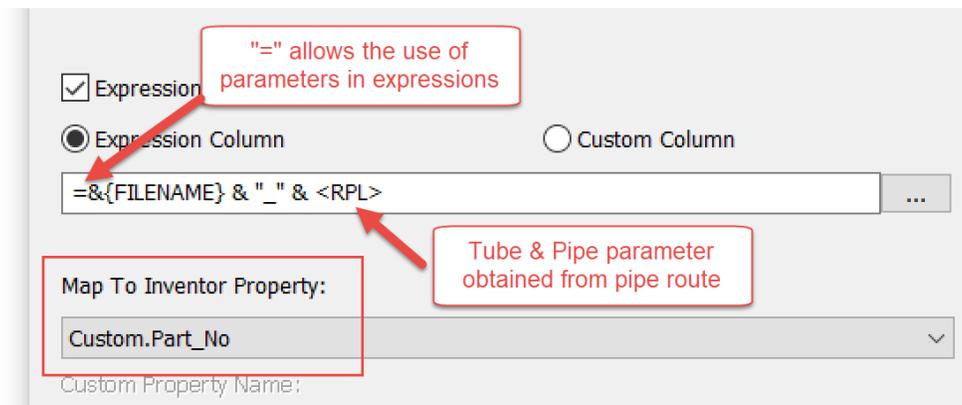
Now that the custom iProperty has been published to the Content Center Family, a new column can be created, custom expressions applied and then mapped to the custom iProperty. Since this component is a pipe component, there are some unique parameters that are created once a pipe is placed in a pipe run. These parameters, which are also exported as custom iProperties, have caused confusion over the years, so I want to take a moment to offer a definition for each of them.

- PL – The exact part length of the tube, hose or pipe based on the piping route segments
- RPL – A corrected hose part length, rounded up to the next desired increment (i.e. rounding up to the nearest 1/2”).
- OPL – This is the exact hose length with the hose fittings taken into account, based on the piping route segments
- ROPL – A corrected hose length with fittings taken into account, rounded up to the next desired increment (i.e. rounding up to the nearest 1/2”)

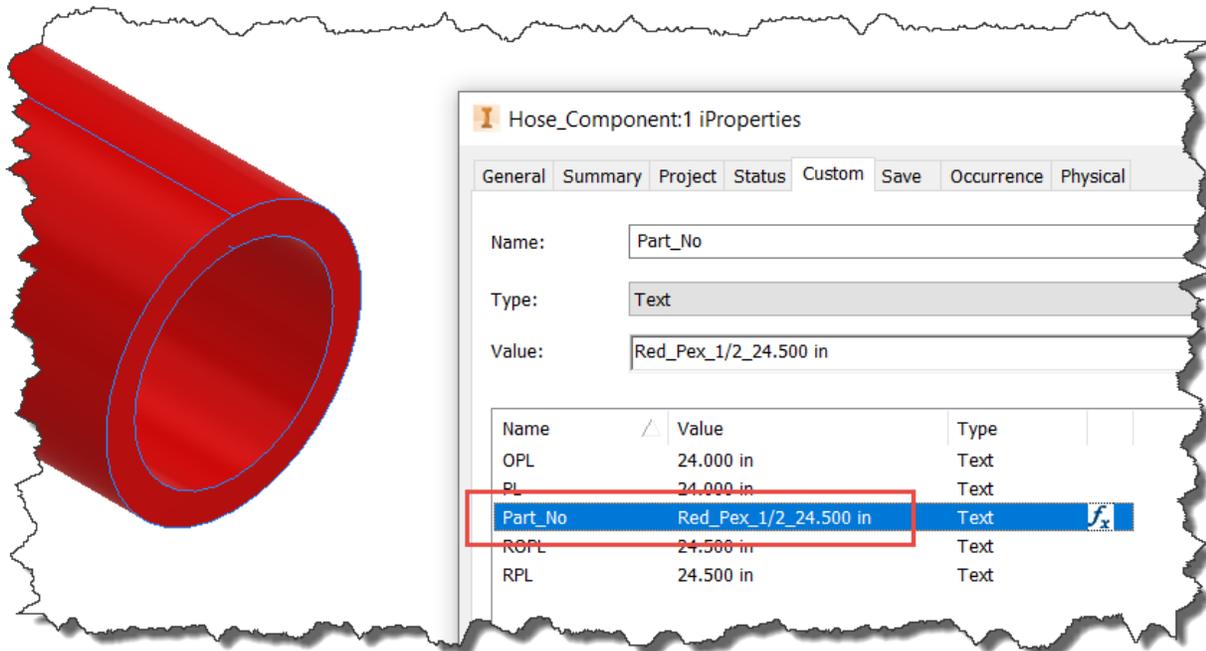
For pipe and formed tube parts, I typically recommend the PL parameter to construct the part number, since those must be exact parts. However, for hoses, I will use the RPL to take advantage of the rounding value taken from the active Tube & Pipe style. To access either of these values, we’ll utilize a different type of equation inside of the custom expression.



Create a new column to house the Part Number information



Populate a custom expression using the format that allows the use of parameters and we'll use the RPL – "Rounded Part Length" for this hose component and be sure to map it to the "Part\_No" iProperty



Custom iProperty populated from the Content Center table after placement in a tube run

Once the custom iProperty is populated and the pipe component is placed in an assembly, we can utilize iLogic functionality to obtain the Part Number results.

### Summary of Content Center Considerations

- Add the “Part\_No” custom iProperty and publish to the Content Center Family
- Create a custom Family column
- Apply the expression utilizing the “<RPL>” parameter
- Map the custom Family column to the “Part\_No” custom iProperty

### iLogic rule to get the job done

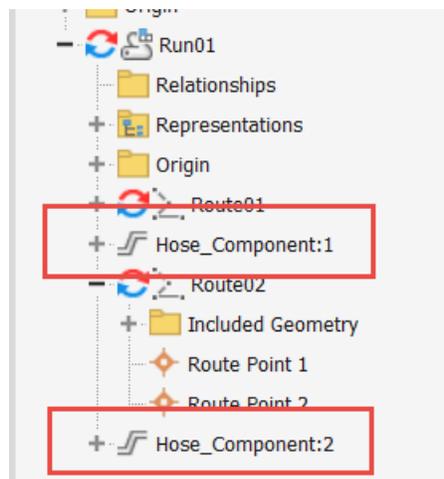
Since there's already a custom iProperty which contains the information that we need, it is a relatively easy task to have an iLogic rule assign the custom iProperty value to the Part Number iProperty. However, there is a wrinkle, we don't want to reassign the Part Number field for all components, only the pipes, formed tubes and hoses placed via the tube and pipe environment.

This isn't really an iLogic class, so we're not going to explore too much theory here, but there are several important items to highlight. Firstly, we'll want to dig into every single subassembly to see if there are any conduit (pipe, formed tube or hose) parts. Next, the tube and pipe environment assigns a generic assembly browser name, which can cause some confusion, so we'll clean that up. Finally, if the rule finds a conduit part, the "Part\_No" iProperty will be copied to the Part Number iProperty. The full rule is available in Appendix C, but we'll examine some of snippets next.

```
'Gain access to all the leaf (part) occurrences in the entire assembly
oComps = ThisDoc.Document.ComponentDefinition.Occurrences
Dim oLeafOccc As ComponentOccurrencesEnumerator
oLeafOccc = oComps.AllLeafOccurrences
```

Search inside of each subassembly for the conduit parts

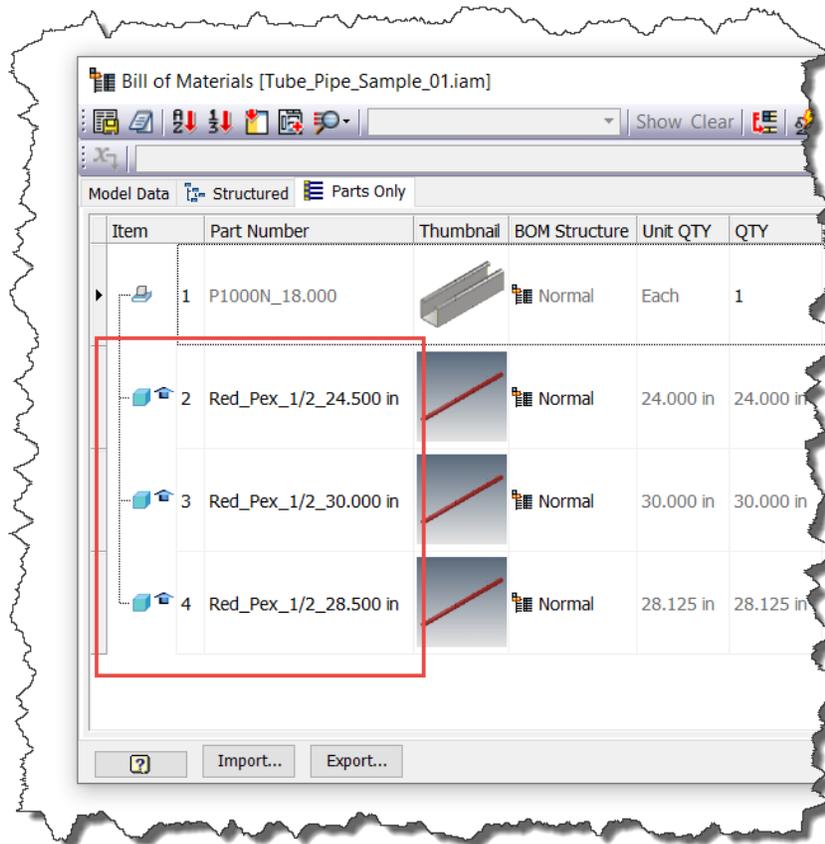
```
Case "pif"
  oComp.Name = "Hose_Component:" & Hose_Count
  Hose_Count = Hose_Count + 1
```



When a conduit component is found, assign a sequential browser name, even across different routes

```
iProperties.Value(oComp.Name,"Project", "Part Number") = iProperties.Value(oComp.Name,"Custom", "Part_No")
MessageBox.Show(oComp.Name & " now has a part number of " & iProperties.Value(oComp.Name, "Project", "Part N
sic"
```

iLogic rule snippet to assign the Part Number iProperty



Parts List results now showing Part Numbers for conduit components

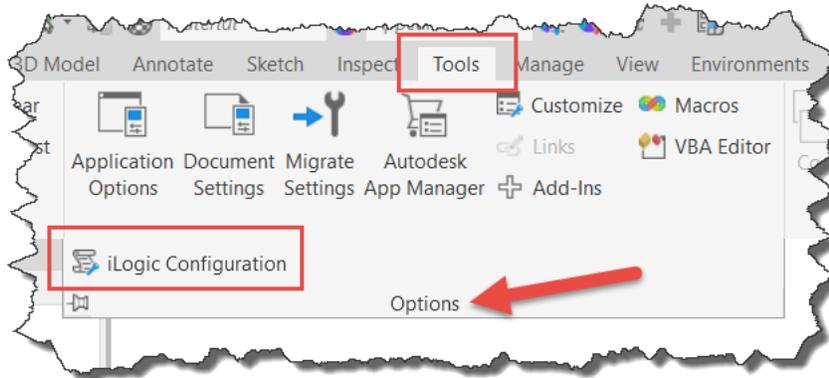
### Summary of iLogic rule

- Search through all the components inside of subassemblies
- Only locate the conduit parts (pipes, formed tubes or hoses)
- Rename the conduit assembly browser nodes to eliminate confusion and improve iLogic code performance
- Copy the value from the “Part\_No” custom iProperty to the Part Number iProperty for every conduit part

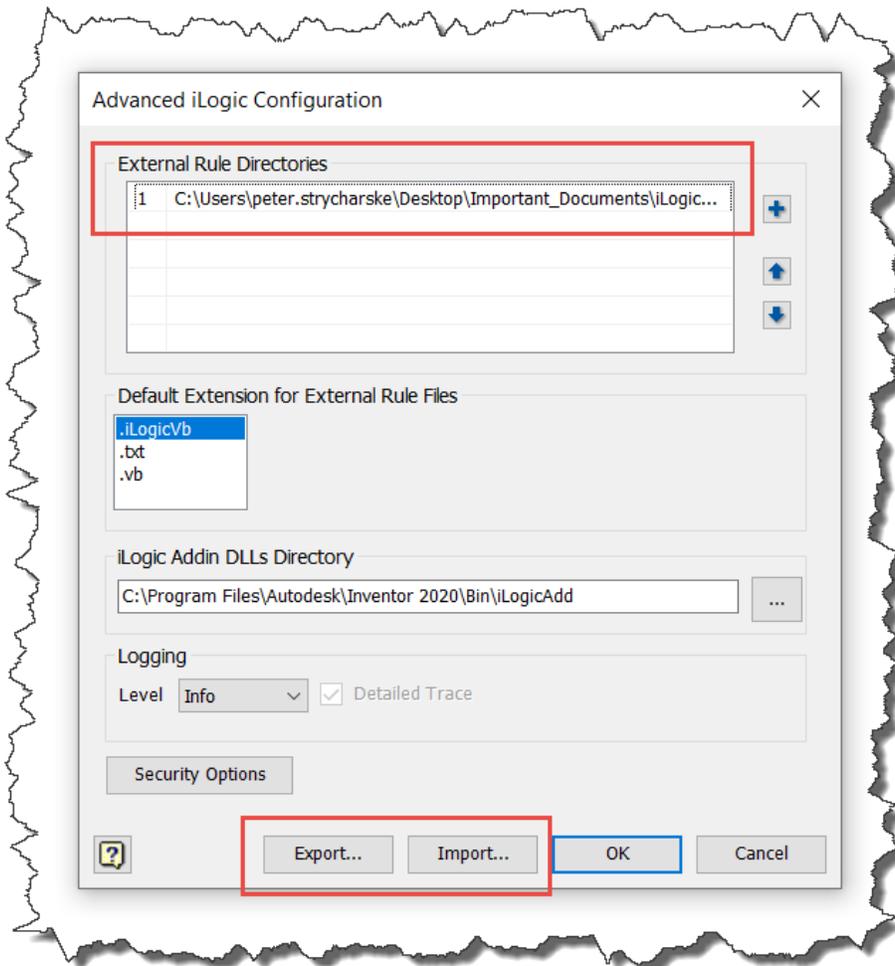
### Setting up and triggering the external iLogic rule

The results of the iLogic rule are exactly what we need to populate the Part Number iProperty, but we can make life easier for the users by implementing this process with an external iLogic rule. External iLogic rules are stored in a central location and do NOT have to be stored in every component. This allows a higher level of consistency across designs by eliminating rule changes in any given file.

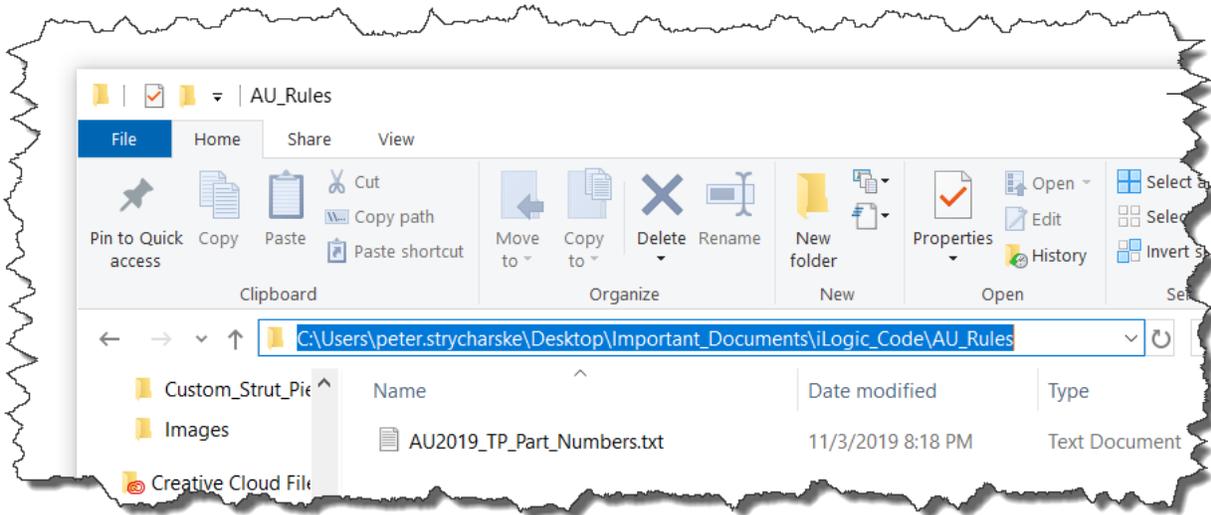
To set up an external iLogic rule, we need to set up a location for iLogic rules and store the iLogic rule as a text file (“.txt”) in that location. There are a couple of simple steps to get this implemented.



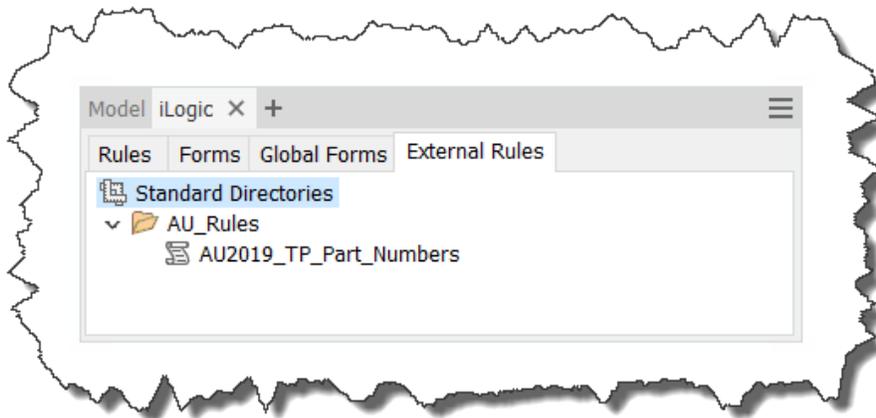
Access the iLogic Configuration tool inside the expanded Options panel



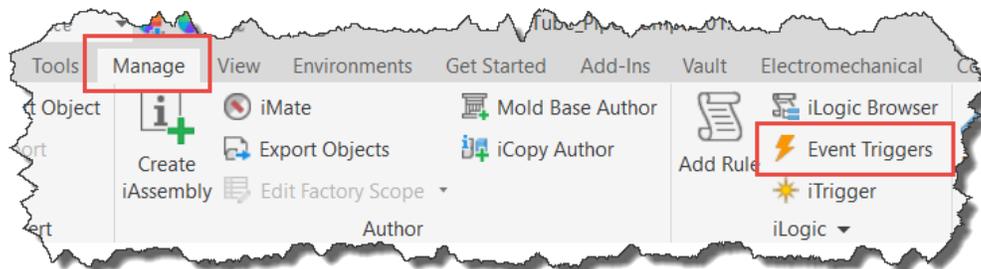
Add and prioritize External Rule Directories and Export / Import these settings to other user's computers



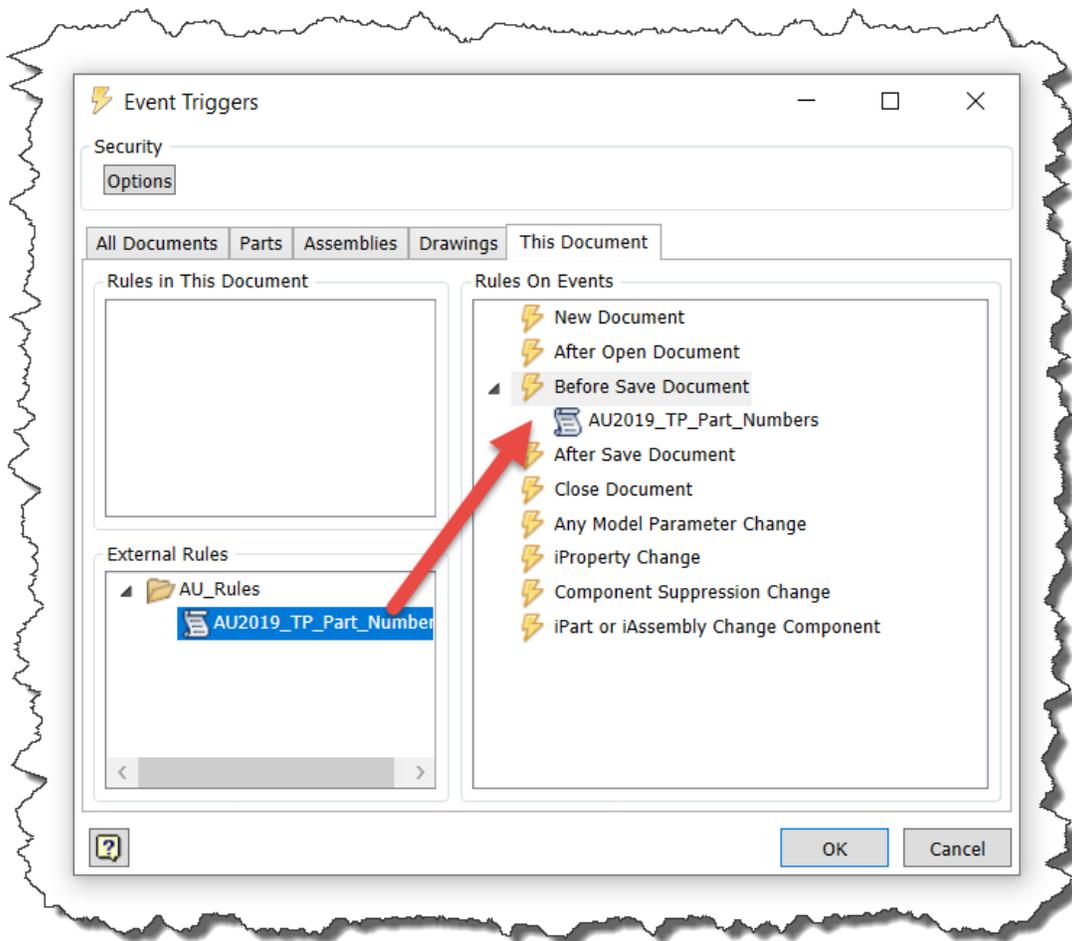
Save iLogic rule text as text files (.TXT) and store them in the external rules location



External iLogic rules referenced inside of Inventor



Within the Assembly template files, access the Event Triggers tool



Typically set iProperty iLogic rules to run “Before Save Document”

By editing the assembly template files and setting external rules to run prior to the save, this ensures that more consistent data is populated into the Parts List.

### Summary of configuring external iLogic rules

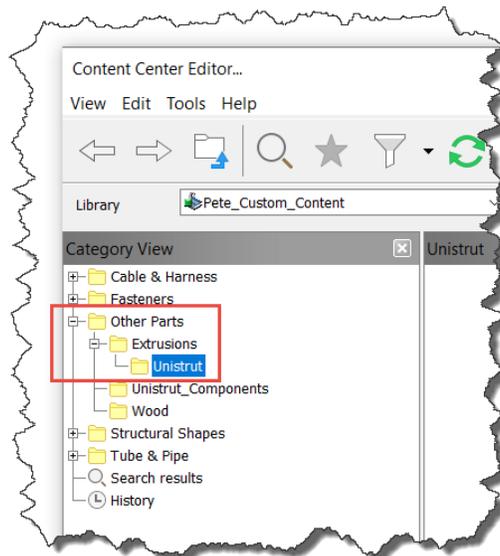
- Set up an external rules location that every user can access
- Export / Import these location settings to every user
- Save iLogic rule text into text files (.TXT) and store them in the external rules location
- Open up the assembly template files and configure the external rule to run before saving the document

## Bonus Topic: A workflow for moving a Content Center family to a different category WITHIN a custom library

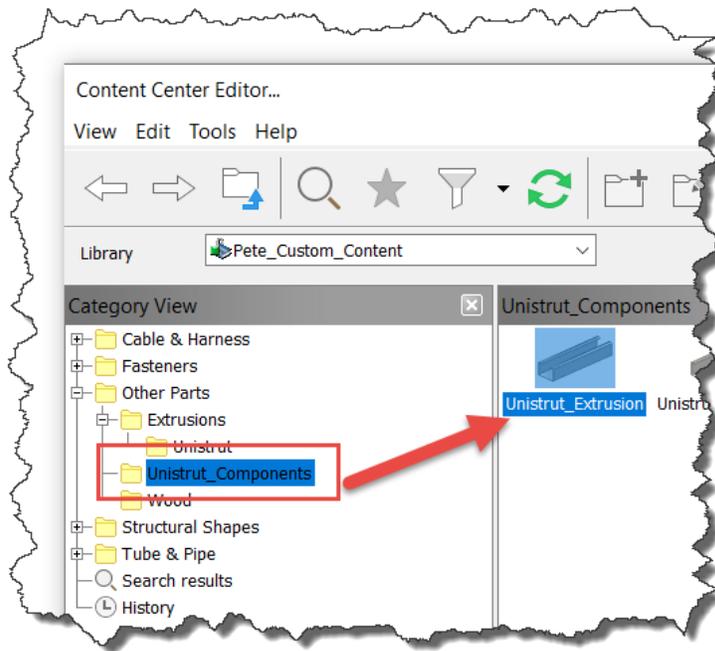
I get asked this question all the time, “how can I move a Content Center Family to a different category, WITHIN the same custom Content Center library?” Frankly, if I had a \$100 bill for every time I’ve been asked that I could probably have used that for my airfare! Okay, so this doesn’t come up all the time, but there are a good number of people who have wondered this and there isn’t really a great answer. The only current workflow that I typically hear about is to republish the component to the Content Center. However, I’d like to show how this could be accomplished using techniques discussed in this report.

### The Workflow

This is actually a twist on the current approach of republishing the desired family, but we’ll utilize what we’ve learned about the External Excel spreadsheets to help accomplish the goal. This works REALLY well within the “Other” category, but I’ve also had success with some “Fastener” categories. The next series of screen shots and captions will walk us through the process.



First create the desired category structure



Find and place a member from the Family to move

AutoSave  Off Unistrut\_Extrusion.xlsx - Excel

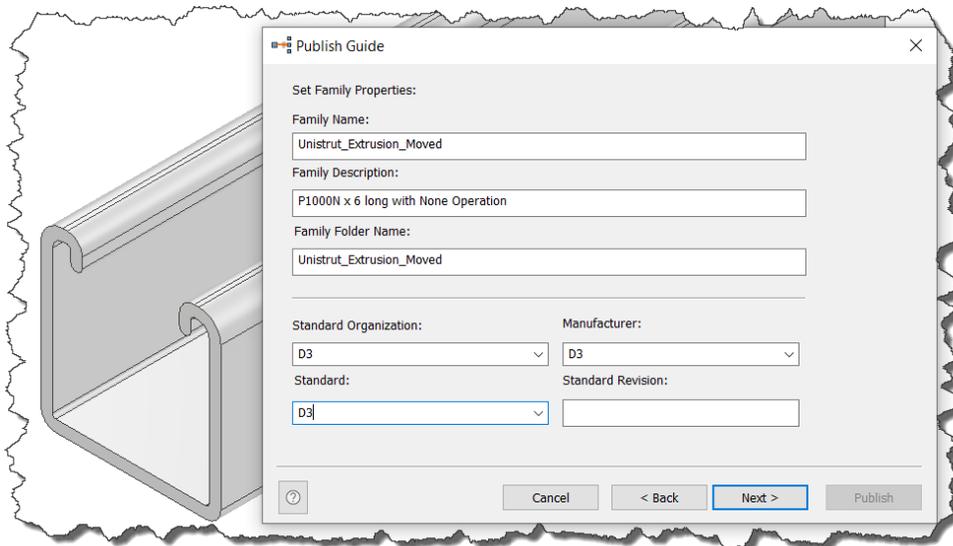
File Home Insert Page Layout Formulas Data Review View Help Acrobat Search

Clipboard Font Alignment Number Styles

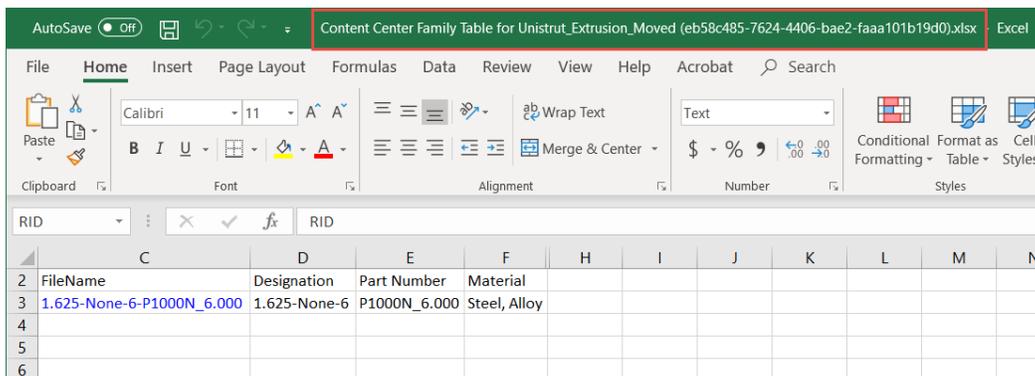
P6 1.625-T\_Slot-6

	C	D	E	F	G	H	I	
2	Member	Part Number [Project]	Channel_Width	Channel_Height	Channel_Thickness	Channel_Open_Gap	OA_Part_Length	T
3	P1000N	P1000N_6.000	1.625	1.625	0.105	0.875	6	t
4	P1000N	P1000N_10.000	1.625	1.625	0.105	0.875	10	t
5	P1000N	P1000N_18.000	1.625	1.625	0.105	0.875	18	t
6	P1000T	P1000T_6.000	1.625	1.625	0.105	0.875	6	f
7	P3300N	P3300N_6.000	1.625	0.875	0.105	0.875	6	t
8	P3300T	P3300T_7.000	1.626	0.875	0.105	0.875	7	f

Edit the Family table via Excel and save to an external directory

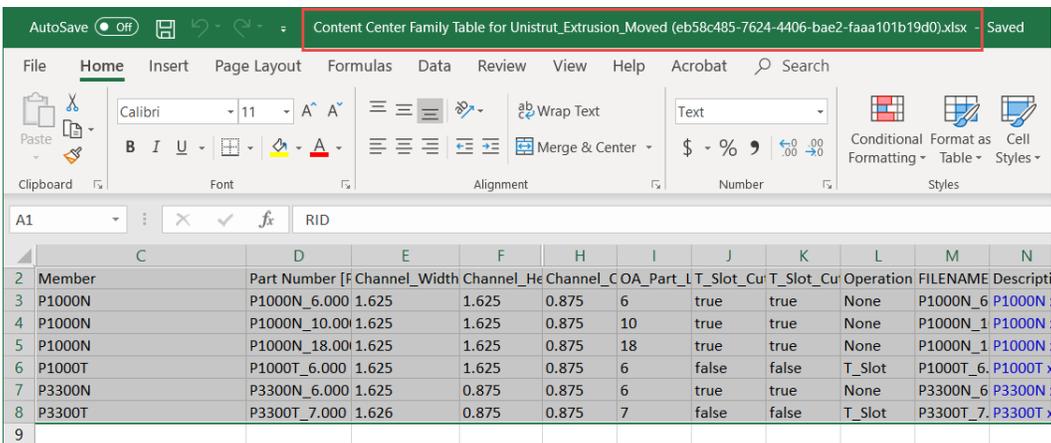


Publish the placed member to the new Family location and rename it slightly



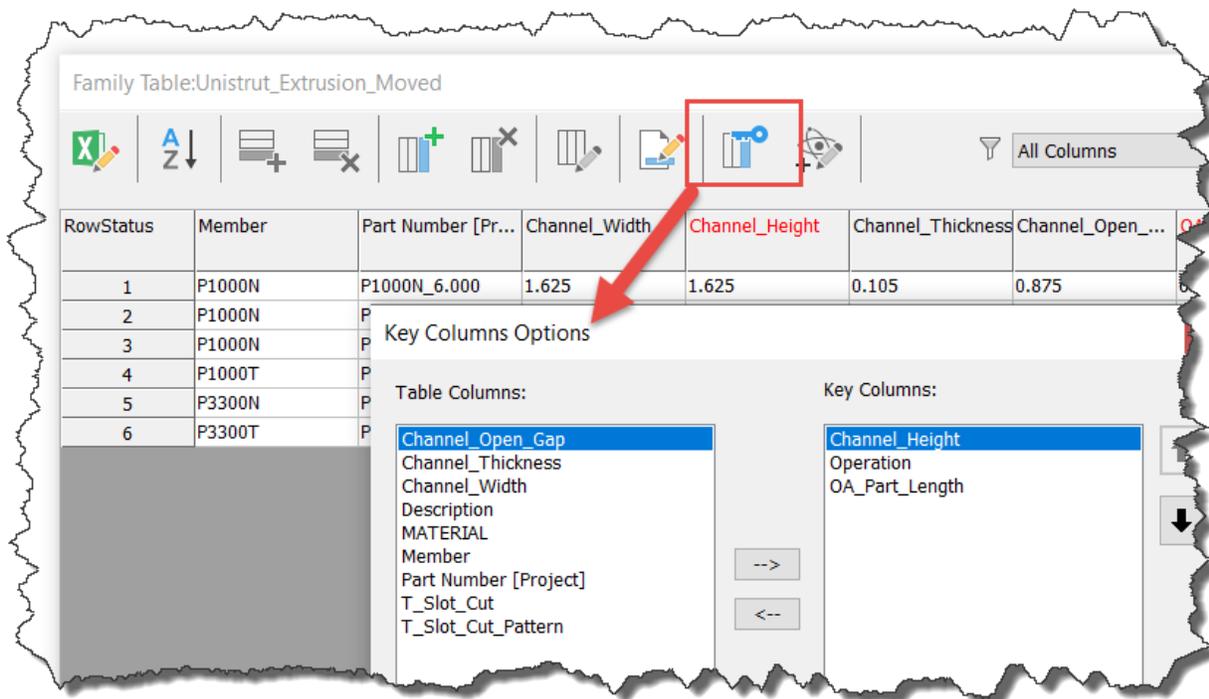
RID	File Name	Designation	Part Number	Material
2	1.625-None-6-P1000N_6.000	1.625-None-6	P1000N_6.000	Steel, Alloy
3				
4				
5				
6				

Edit the newly published Family table via Excel



Member	Part Number	[FChannel_Width	Channel_He	Channel_COA	Part_L	T_Slot_Cu	T_Slot_Cu	Operation	FILENAME	Descripti
P1000N	P1000N_6.000	1.625	1.625	0.875	6	true	true	None	P1000N_6	P1000N :
P1000N	P1000N_10.000	1.625	1.625	0.875	10	true	true	None	P1000N_1	P1000N :
P1000N	P1000N_18.000	1.625	1.625	0.875	18	true	true	None	P1000N_1	P1000N :
P1000T	P1000T_6.000	1.625	1.625	0.875	6	false	false	T_Slot	P1000T_6	P1000T :
P3300N	P3300N_6.000	1.625	0.875	0.875	6	true	true	None	P3300N_6	P3300N :
P3300T	P3300T_7.000	1.626	0.875	0.875	7	false	false	T_Slot	P3300T_7	P3300T :

Open the external Excel sheet from the same session, copy and paste the data into the original Family table via an Excel edit



Edit the Family table column properties and key columns as needed

Place a member from the new family and test things out before eliminating the original family. It is NEVER a good idea to release new untested tools on your unsuspecting colleagues... There are also some important considerations to ensure that this process works smoothly, which I'll list below. Use this as a workflow to “move” a family from one category to another, within the same custom library!

### Important Considerations for a Content Center Family Move

- Some of the categories have very specific Family properties, which makes relocating from that category more complex. For example, the Category Properties of a Hex Head Bolt differs from a Round Head Bolt.
- If your original family is based on an iPart table, typically for controlling features, then you must republish from the original iPart to the new location, otherwise the features will not be properly controlled in the “moved” Family.
  - In general, it is a good idea to save a copy of every base component that is used to publish to the Content Center, iPart or otherwise. This way if a one off design or a republish is desired, the saved version can be a great launch point.

Category Properties

General Parameters

Parameter Name	Data Type	Units	Category Path	Mapping
Grip Length	Real	cm	Fasteners->...	Required
Head Height	Real	cm	Fasteners->...	Required
Property Class	String		Fasteners->...	Optional
Thread Length	Real	cm	Fasteners->...	Required
Shank Diameter	Real	cm	Fasteners->...	Optional
Head Diameter	Real	cm	Fasteners->...	Required
Nominal Diameter Designation	String		Fasteners->...	Optional

Category Properties

General Parameters

Parameter Name	Data Type	Units	Category Path	Mapping
Grip Length	Real	cm	Fasteners->...	Required
Width Across Flats	Real	cm	Fasteners->...	Optional
Head Height	Real	cm	Fasteners->...	Required
Thread Length	Real	cm	Fasteners->...	Required
Width Across Corners	Real	cm	Fasteners->...	Optional
Shank Diameter	Real	cm	Fasteners->...	Optional
Diameter of Contact Area	Real	cm	Fasteners->...	Optional
Nominal Diameter Designation	String		Fasteners->...	Optional
Nominal Length	Real	cm	Fasteners->...	Required

A comparison of the Round Head (Above) and Hex Head (Below) category properties

## Appendix A: Weblinks to technical articles

Link to advanced iPart table editing in Excel blog post

<https://go.d3tech.net/inventor-blog/inventor-tips-and-tricks-utilizing-excel-for-advanced-ipart-table-editing>

Link to Excel versus relational database performance article

<https://news.codecademy.com/excel-to-sql-why-switch/>

## **Appendix B: Weblinks to demonstration videos**

Link to Placed Shuttleworth iPart

<https://autode.sk/303vvjY>

Link to Placed Shuttleworth Content Center Family

<https://autode.sk/31s0a80>

Link to Publish iPart Custom Column to Content Center Family

<https://autode.sk/32x4m60>

Link to Custom Expression Column in Content Center Family

<https://autode.sk/2Q6mzEY>

Link to Utilize an External Excel File to Manage Content Center Family Data

<https://autode.sk/2Q8jqVe>

Link to Tube & Pipe Conduit Part Numbering Workflow

<https://autode.sk/2Q96Y7q>

Link to Bonus Top – Workflow to Move a Content Center Family

<https://autode.sk/2Cw8SHm>

## Appendix C: iLogic rule text

### Part Number Rule (with Content Center Component)

```
Dim Part_Length_Des As String
Length = CStr(OA_Part_Length)
String_Length = Len(Length)

Select Case String_Length
    Case 1
        Part_Length_Des = "000" & Length
    Case 2
        Part_Length_Des = "00" & Length
    Case 3
        Part_Length_Des = "0" & Length
    Case 4
        Part_Length_Des = Length
End Select

MessageBox.Show(Part_Length_Des, "iLogic Rule Setting Part Length String")

iProperties.Value("Project", "Part Number") = iProperties.Value("Custom",
"Comp_Des") & "_" & Part_Length_Des

MessageBox.Show(iProperties.Value("Project", "Part Number"), "iLogic Rule
Setting Part Number")
```

## Tube & Pipe Part Numbers Rule

```

Dim oComp As ComponentOccurrence
Dim oComps As ComponentOccurrences

'Gain access to all the leaf (part) occurrences in the entire assembly
oComps = ThisDoc.Document.ComponentDefinition.Occurrences
Dim oLeafOccs As ComponentOccurrencesEnumerator
oLeafOccs = oComps.AllLeafOccurrences

Dim oCompDef As ComponentDefinition

''Set up Part counters
Dim PipeCount As Integer
Dim Tube_Count As Integer
Dim Hose_Count As Integer

Pipe_Count = 1
Tube_Count = 1
Hose_Count = 1

'Loop through each component in the assembly
For Each oComp In oLeafOccs
    oCompDef = oComp.Definition

    'Set the Full File Name Variable
    Full_Current_File_Name = oCompDef.Document.FullFileName
    'Get the location for the last \ in the full file name and path
    sIndex = Strings.InStrRev(Full_Current_File_Name, "\")
    'Set the file name by subtracing the number of charactes from the ful
    path by the sIndex number
    File_Name = Right(Full_Current_File_Name, Len(Full_Current_File_Name)
    -sIndex)

    'Logic for what happens with the different types of components
    Select Case Left(File_Name, 3)
        Case "pip"
            oComp.Name = "Pipe_Component:" & Pipe_Count
            Pipe_Count = Pipe_Count + 1

            iProperties.Value(oComp.Name,"Project", "Part
            Number") = iProperties.Value(oComp.Name,"Custom", "Part_No")
            MessageBox.Show(oComp.Name & " now has a part
            number of " & iProperties.Value(oComp.Name, "Project", "Part Number"), "New
            Part Number")

        Case "pif"
            oComp.Name = "Hose_Component:" & Hose_Count
            Hose_Count = Hose_Count + 1

            iProperties.Value(oComp.Name,"Project", "Part
            Number") = iProperties.Value(oComp.Name,"Custom", "Part_No")
    
```

```
        MessageBox.Show(oComp.Name & " now has a part
number of " & iProperties.Value(oComp.Name, "Project", "Part Number"), "New
Part Number")
        Case "pis"
            oComp.Name = "Tube_Component:" & Tube_Count
            Tube_Count = Tube_Count + 1

            iProperties.Value(oComp.Name, "Project", "Part
Number") = iProperties.Value(oComp.Name, "Custom", "Part_No")
            MessageBox.Show(oComp.Name & " now has a part
number of " & iProperties.Value(oComp.Name, "Project", "Part Number"), "New
Part Number")
        Case Else
            'Put code in here if you want something to happen to
all the other parts
        End Select
    Next
```