



Finding the Logic in iLogic™: Hands On

Stan Wile – Avatech Solutions

MA222-L

About the Speaker:

Diversified and experienced, Stan is a valuable team member with Avatech Solutions. As an accomplished and proven engineer, he has significant experience in all aspects of 3D conceptual design.

Leading former employers, Stan was instrumental in implementing the switch from AutoCAD® to Inventor®. He also implemented Vault to solve data management issues with multiple engineers working on the same projects. He has written user manuals for all drafting procedures including data management with Vault and an entire plant manual for ISO 9000 certification.

Stan has quickly become one of Avatech's "go-to AEs" for the more complex implementations. Some of these include Miller Weldmaster, Belvac, and Atlas Copco. He is also Avatech's East Coast AutoCAD Electrical expert. Stan is an adept and experienced instructor who teaches all Inventor course offerings along with AutoCAD and AutoCAD Electrical.

Stan.Wile@avatech.com

This hands-on approach to iLogic will deliver an in-depth look at the exciting functionality delivered with the iLogic add-on to Autodesk® Inventor®. We will focus on automating tedious design tasks with a rules-based design workflow. The exposure to this automation process will drive innovation in your environment. You will be able to put this skill to use in your workplace immediately. Finding the logic behind iLogic is critical to getting the most “bang for your buck” with Autodesk Inventor. For a sample dataset, I chose a simple Post Base for a Strut Channel. This is something simple yet also has many variables. Because this is a simple component to model, we can jump into iLogic right away.

Post Bases for Strut Channel

Create a solid foundation when you mount these bases to the floor to support upright 1 1/2" x 1 3/8" strut channel (not included). Mounting holes are spaced on 4 1/4" centers (unless noted). Mounting bolts are not included. *Note:* Base height does not include plate thickness. **Steel** and **stainless steel** bases have four 3/4" dia. mounting holes (unless noted). **Fiberglass** bases have four 7/8" dia. mounting holes. Color is gray.

To Order: For choose-a-color steel, please specify color: black, red, white, or yellow.

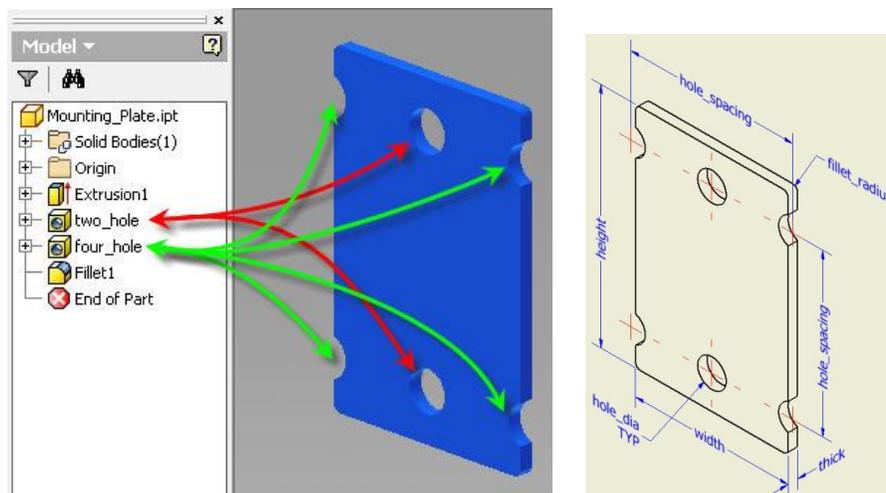
Material	Plate Thick.	Base Ht.	SINGLE-CHANNEL BASES			DOUBLE-CHANNEL BASES		
			4"x6" Plate Standard Post	6"x6" Plate Standard Post	6"x6" Plate Offset Post	6"x6" Plate Standard Post	6"x6" Plate Offset Post	
Zinc-Plated Steel	1/4"	3 1/2"	33145T53▲	33145T32	33145T19	33145T28	33145T44	
Green-Painted Steel	1/4"	3 1/2"	33145T51▲	33145T31	33145T21	33145T29	33145T45	
Choose-a-Color Steel	1/4"	3 1/2"	33145T11	33145T16	33145T22	33145T36	33145T46	
Galvanized Steel	1/4"	3 1/2"	33145T13	33145T18	33145T24	33145T38	33145T48	
Type 304 Stainless Steel	1/4"	3 1/2"	33145T54▲	33145T33	33145T25	33145T39	33145T49	
Type 316 Stainless Steel	1/4"	3 1/2"	33145T14	33145T34	33145T26	33145T42	33145T55	
Fiberglass	3/8"	3"	33145T41▲			33145T43▲		

▲ Have two 1 1/8" dia. mounting holes spaced on 4" centers. ➤ Mounting holes are spaced on 4" centers. ▲ Base plate is 5" x 5"; mounting holes are spaced on 3 3/4" centers. ♣ Base plate is 6 1/2" x 5"; mounting holes are spaced on 5 7/8" and 3 1/8" centers.

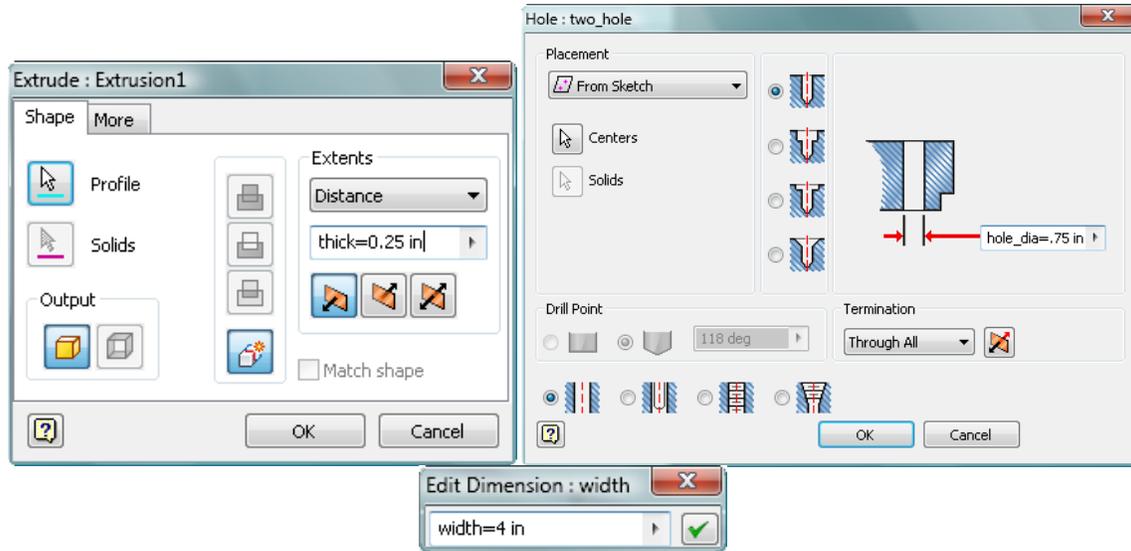
1512 **McMASTER-CARR**

Start Inventor 2010 and open the file called **mounting_plate_no_rules.ipt**. Use the Save As command to save this document as a new file, named “mounting_plate.ipt”. This is the file that we will use for the first part of this training.

Before we start building logimetric intelligence in this part, we need to decide what we are trying to accomplish and understand how the design intent of the part up to this point. We are focusing on the mounting plate of the post base assembly shown above. Since the bolt pattern can be either two or four holes, they are modeled as individual features. This allows us to control the visibility on each feature as needed. Every parameter that may change has a specific name to make it easier to identify driving parameters later.

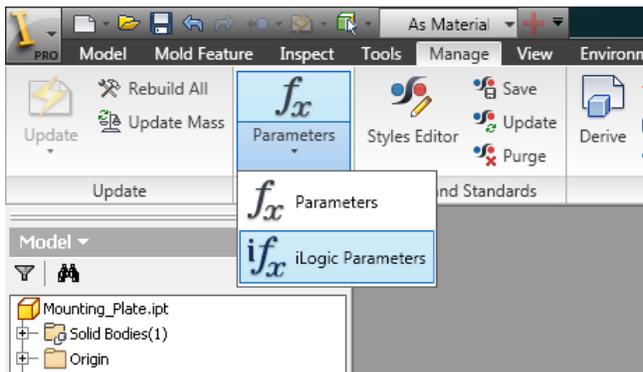


With Inventor 2010, you can now name a parameter right inside any dialog box. The format is *parameter_name=value*. Take advantage of this in any dialog box that accepts parameter value input.

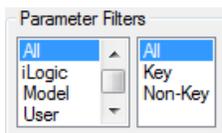


OPEN THE ILOGIC PARAMETER EDITOR

Switch to the Manage tab of the Inventor ribbon. On the Parameters panel, locate the Parameters button. Expand the dropdown menu at the bottom of this button, and notice the iLogic Parameters command. Activate this command to bring up the iLogic Parameter Editor.



In the Parameter Filters box at the bottom left of the Parameter Editor window, select the “All” option in both lists. This will display both standard and iLogic parameters associated with the bracket model. We will use this filter in more detail a little later.



We will create three iLogic parameters for this model. Each parameter will be utilized to drive a certain characteristic of the model

“struts”: Determines “Single” or “Double” Channels

“fillet”: Determines the existence of the fillet

“material”: Determines available materials.

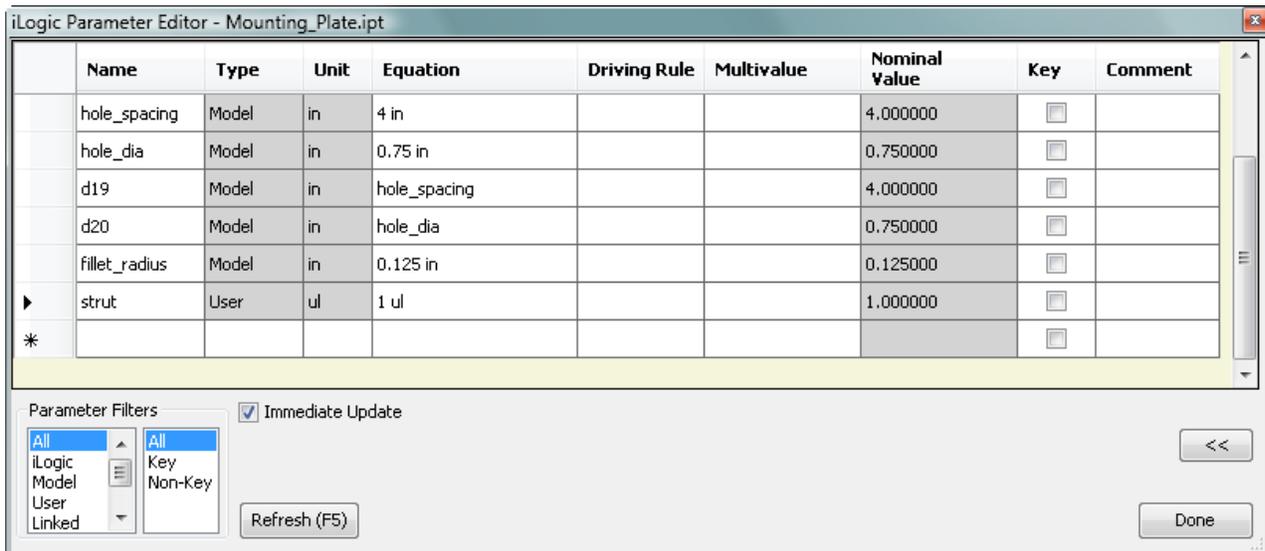
Parameter types explained

- **User parameters** use the standard Inventor unit types.
- **String parameters** hold text values.
- **Boolean parameters** are True or False statements.
- **Multi-Value Parameters**
 - You can right-click on any parameter (in the Equation or Nominal Value column) to create a multi-value list. A dialog box will prompt you for a list of values and generate a pull-down list of parameter values to choose.

Note: iLogic parameters do not support equations in the parameter editor.

CREATE THE “STRUT” PARAMETER

Type “**strut**” in the “Name” field and select the “**User**” option for the struts parameter. Change the unit to “**ul**” and type “**1**” in the “Equation” field.

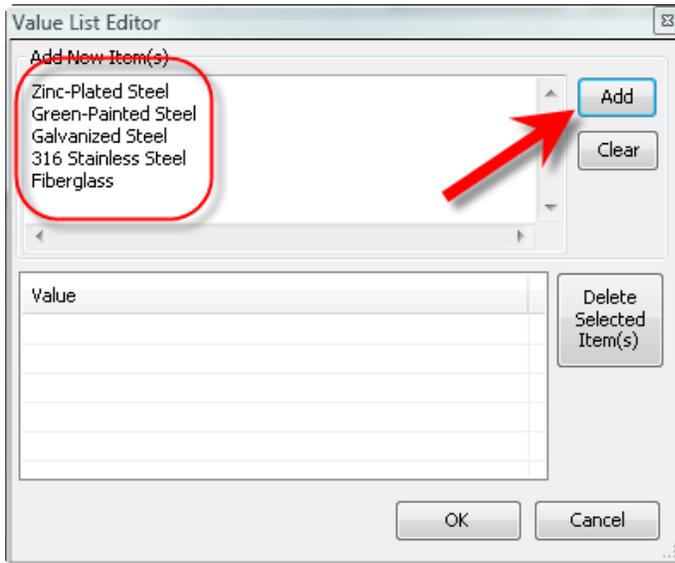


CREATE THE “MATERIAL” PARAMETER

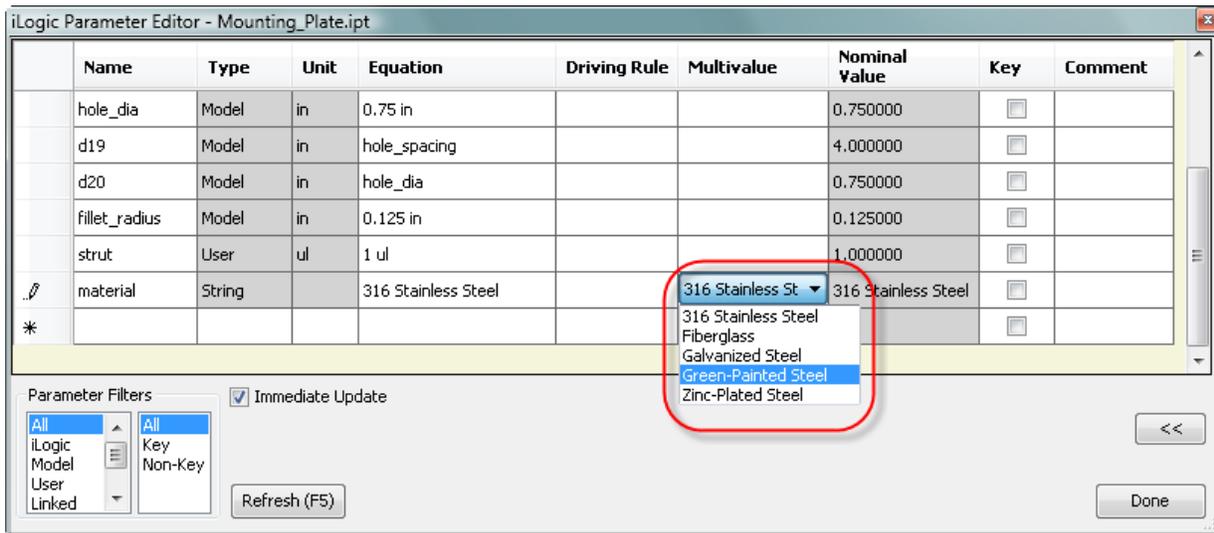
Enter “**material**” as the next iLogic parameter. Choose “**String**” as the parameter type. Right click on the “Multivalue” column and make this a multi-value parameter. Add the materials shown below (copy and paste from this document). Make “Green-Painted Steel” the current value.

- Zinc-Plated Steel**
- Green-Painted Steel**
- Galvanized Steel**
- 316 Stainless Steel**
- Fiberglass**

Press the Add button and then OK.



Now all material values are available from the multivalue drop down.

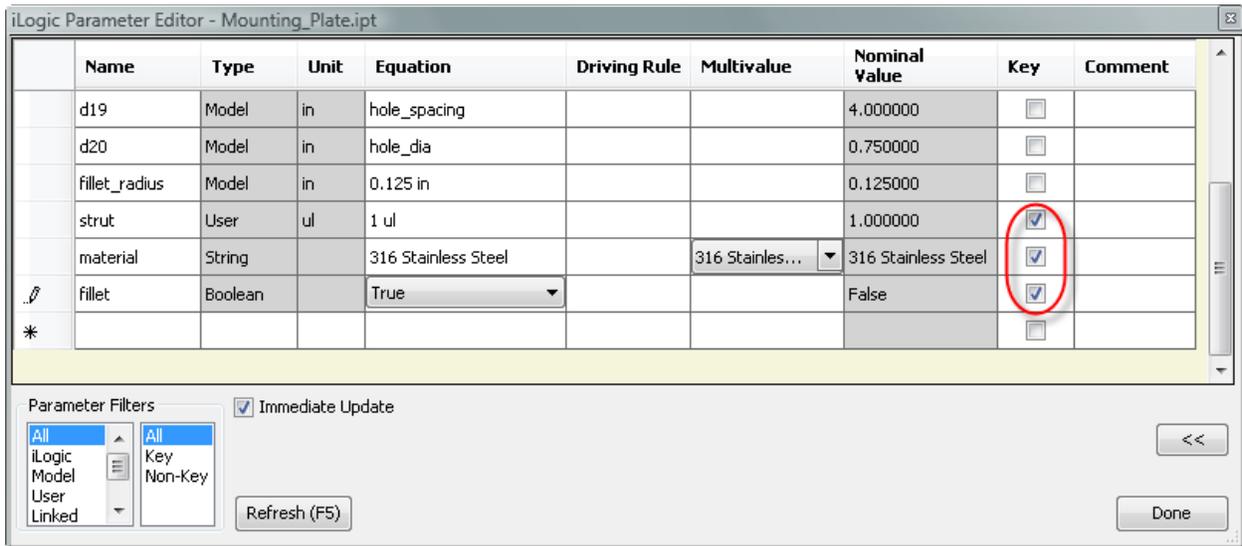


CREATE THE “FILLET” PARAMETER

Add one more parameter named “fillet”. Choose “Boolean” from the “Type” drop down. Click in any other cell and notice a new drop down in the “Equation” cell with “True” and “False” as options. Set the value to “True”

KEY PARAMETERS AND FILTERS

Make sure there is a check in the “Key” boxes for the three new parameters.



It would not be unusually to have dozens and dozens of parameters on large or complex models. The “Key” check box is an optional flag to help you sort important or critical parameters. Use the filters (at the bottom of the dialog) to see only key parameters per this setting.

Try the different options in the parameter filter fields. Notice how the list of displayed parameters updates accordingly. The following table describes the results for the various selections in the Parameters Filter fields.

Left Field Selection	Right Field Selection	Result
All	All	All parameters shown
All	Key	All Key Parameters shown
All	Non-Key	All Non-Key Parameters shown
Model	All	Shows all Model Parameters
Model	Key	Shows all Key Model Parameters

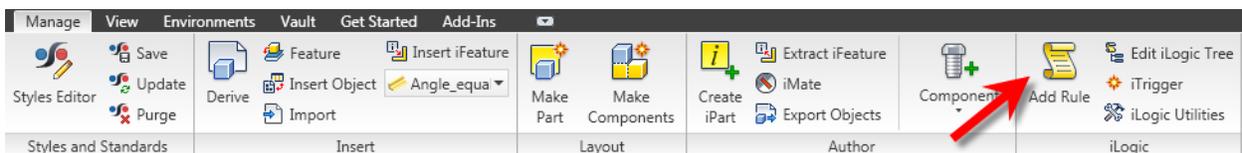
This part has 12 parameters associated with it. A complex part or assembly of parts may have hundreds of parameters. By strategically designating Key parameters, it is possible to find relevant parameters much more easily by filtering the list.

CREATING RULES

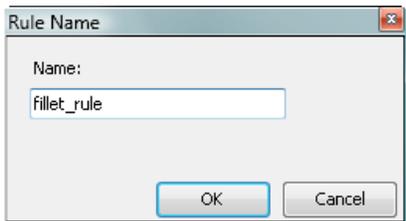
With the necessary parameters in place, we can now add our logic to the model, in the form of rules. Rules describe various actions, including setting parameter values, activating or suppressing features, and many more.

RULE #1 – FEATURE SUPPRESSION

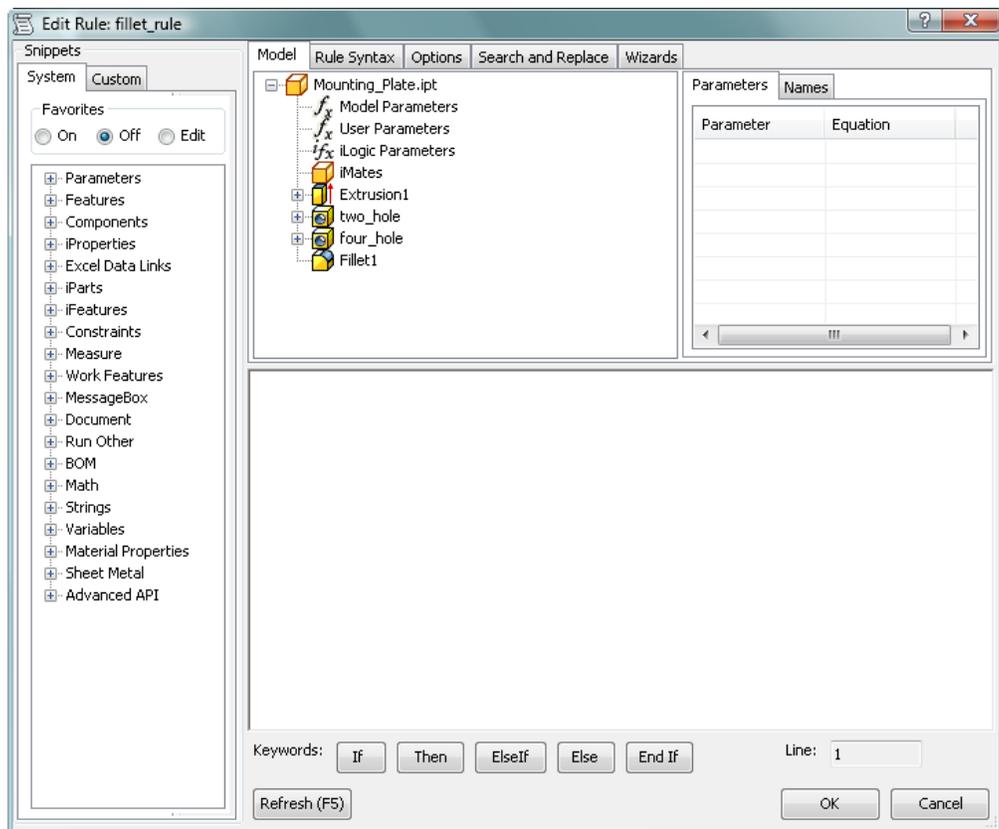
To create our first rule, click on the “Add Rule” icon on the iLogic panel of the Manage ribbon tab.



This opens up the Rule Name dialog. Name the rule “fillet_rule”. (We can change the name of a rule at anytime later on.



Clicking OK opens the iLogic Rule editor dialog.

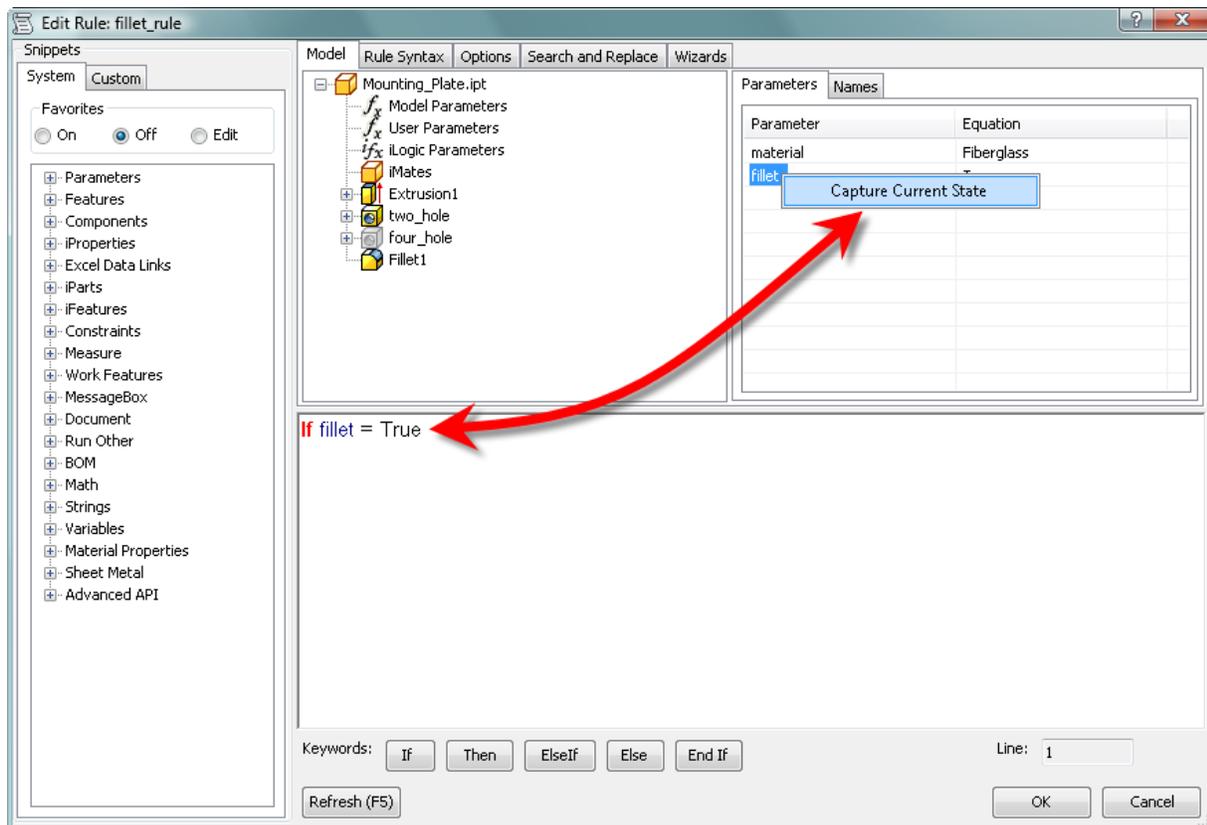


There are several tabs at the top of this dialog box. When the “Model” tab is selected, the model browser and parameter lists are shown. Clicking on the different parameter headings displays the parameter names and values in the list box. You can double-click a parameter name and it will copy to the Rule Window. Keywords can be typed or selected from the common keywords buttons at the bottom of the Rule Window.

Our fillet rule will simply turn on and off the suppression of the Fillet1 feature.

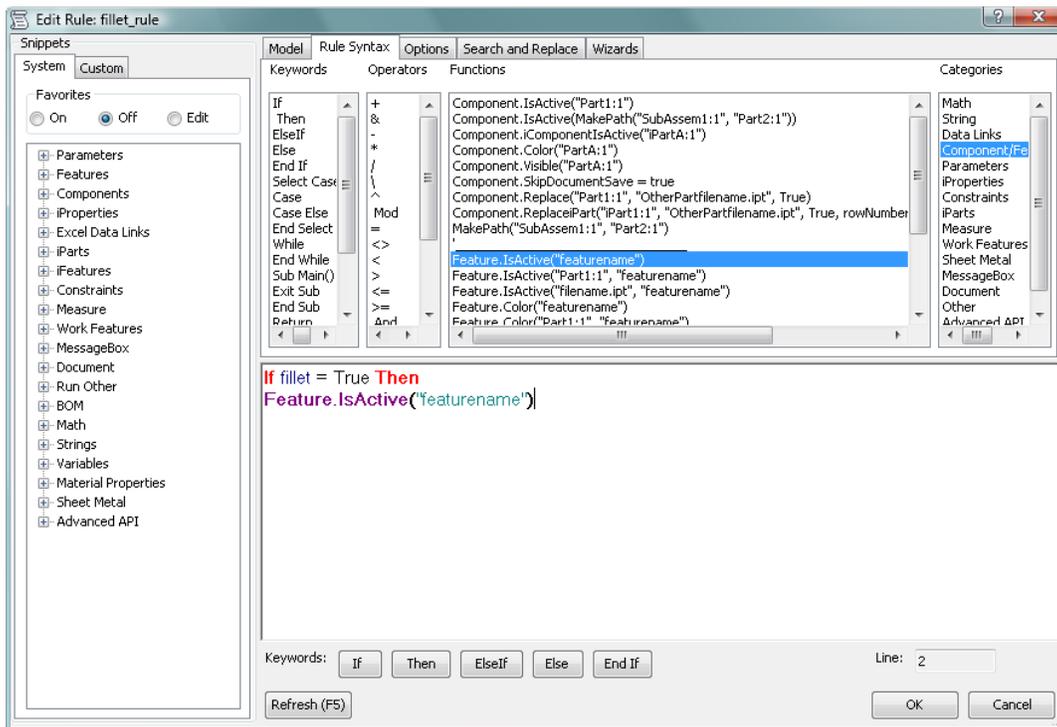
Start by typing “**If**”, or click the “If” button, and then enter a **space**. The text will turn bold red (keywords are always bold red). This color-coding makes rules much easier to read and helps identify any mistyped information.

Select “iLogic Parameters” from under the Model tab and then double-click our “fillet” parameter from the Parameters tab (double clicking a parameter copies it to the rules window) then type “= True”. (Alternatively, you may type the parameter name but be aware that it is always case sensitive).

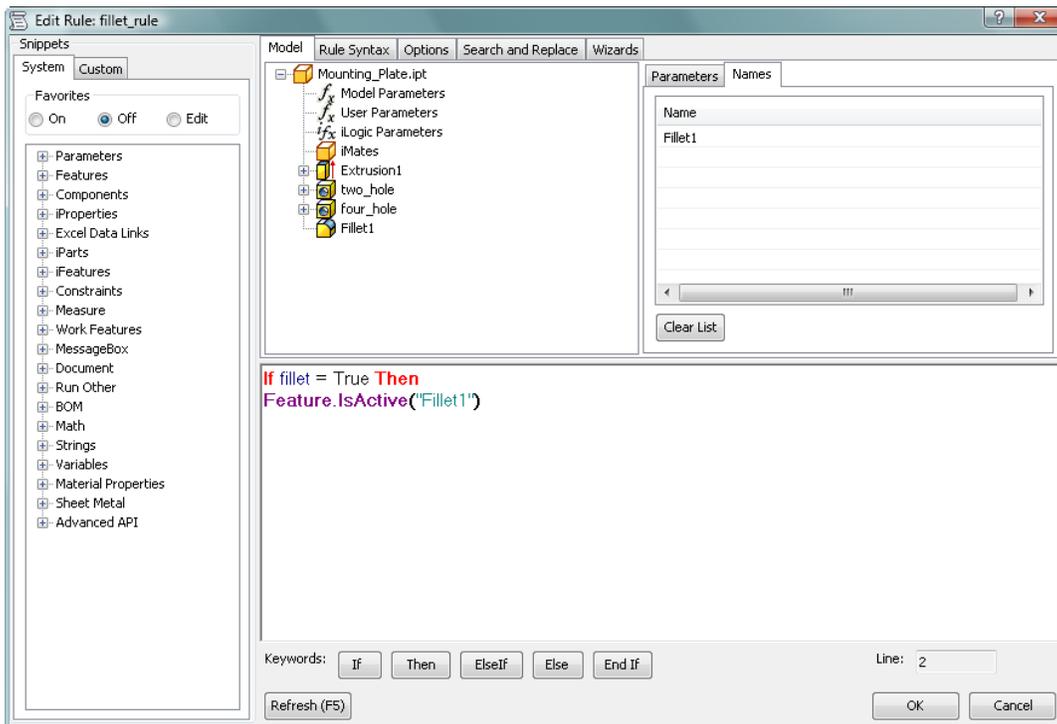


Finish this statement by typing “**Then**” and press **enter** (or click the “Then” button).

Now we will tell iLogic to make Fillet1 active if the parameter is “True”. Click the “Rules Syntax” tab. Choose “Component/Feature” from the “Categories” cell. Then double click “Feature.IsActive(“featurename”)”

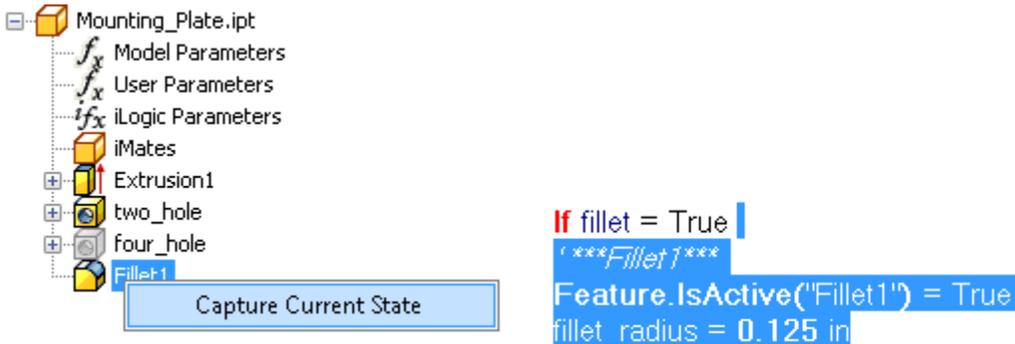


Highlight **“featurename”** from the rules window (leave the quotation marks unselected). Switch back to the “Model” tab and click on “Fillet1” then change to the “Names” tab to the right. Double click on the name **“Fillet1”** and it will overwrite the highlighted text.



Type “= True” behind the last line and then press the “Else” button.

Note: You can also right click on the features feature and choose “Capture Current State” to capture additional information about the feature all at once. See below.



Now highlight and copy (by right clicking and choosing Copy or by pressing Ctrl+C) the second row and paste (by right clicking and choosing Paste or by pressing Ctrl+V) it to the line below the “Else”. Change the “True” value to “False” and type “End If” on the last line. The completed rule should look like this.

```

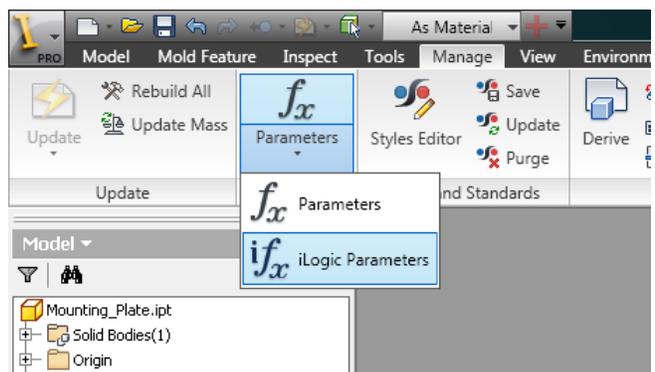
If fillet = True Then
Feature.IsActive("Fillet1") = True
Else
Feature.IsActive("Fillet1") = False
End If
    
```

Click OK in the bottom right of the dialog. If there are no mistakes in the code, the Rule Editor will close without an error message. Save the file (some rules become quite complex and saving before testing the rule is a good habit).

Testing the rule

Open up the iLogic Parameters again

Change the parameter “fillet” from “True” to “False” and note the immediate update on the model. Close the dialog when done. The actual status of the fillet does not matter at this point.



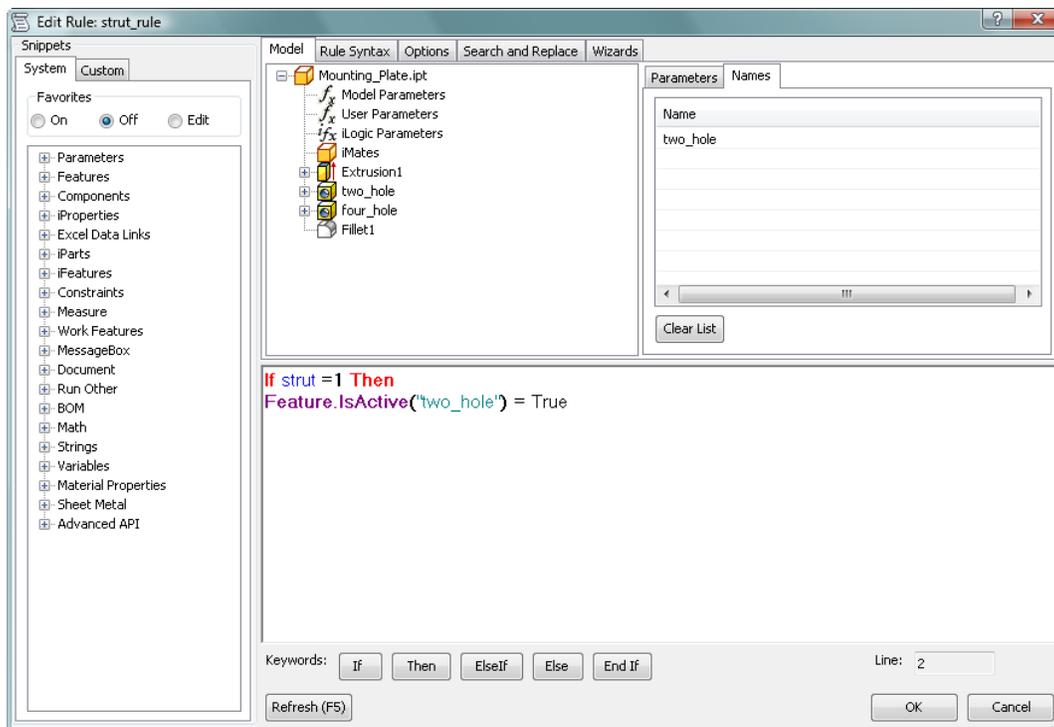
	strut	User	ui	1 ui
	material	String		316 Stainless Steel
	fillet	Boolean		False
				True
				False

RULE #2 – FEATURE SUPPRESSION TOGGLE

Select “Add Rule” and create a new rule named “**strut_rule**”.

This rule will tell the mounting plate whether it is a single or double strut base. When the parameter named strut is set to “1” the base is to have two holes. When set to “2” it will have four holes and be two inches wider.

Start again with an **If** followed by the parameter name **strut**. Remember you can select it from the list and double click it to push it down to the rule window. End the line with “= 1”. On the second line push down **Feature.IsActive** from the “Rule Syntax”, tab again. Highlight “**featurename**” between the quotation marks, and push down the feature name “**two_hole**”. This can be typed in manually or pushed down from the “Names” tab. Remember, press the “Model” tab and click on the feature named “two_hole”.



This code turns on the “two_hole” feature but “four_hole” should be suppressed for now. Copy and paste the second row of text and change “two_hole” to “**four_hole**” and “True” to “**False**”. The completed line should look like this.

```
If strut =1 Then
Feature.IsActive("two_hole") = True
Feature.IsActive("four_hole") = False
```

Press the **ElseIf** button then repeat the first three lines swapping the 1 for a 2 and toggling the True and False text. Press the **End If** button.

Completed code is on the next page.

```

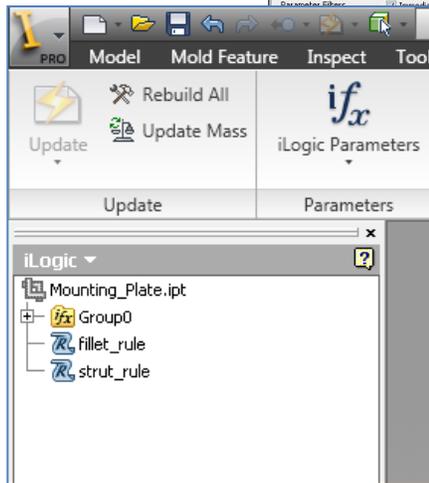
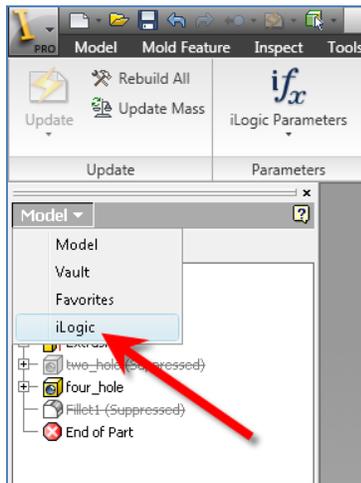
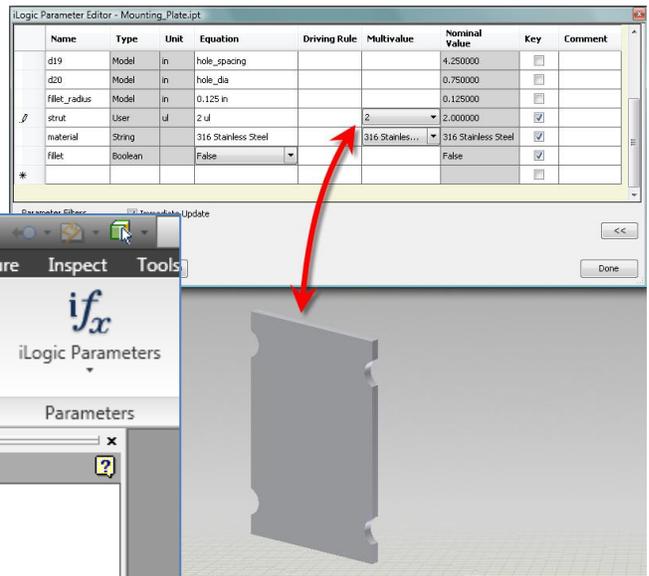
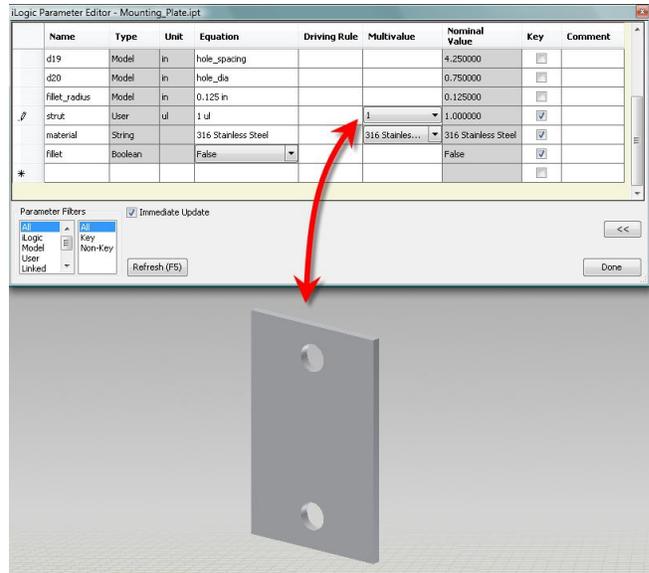
If strut =1 Then
Feature.IsActive("two_hole") = True
Feature.IsActive("four_hole") = False
Elseif strut =2 Then
Feature.IsActive("two_hole") = False
Feature.IsActive("four_hole") = True
End If
    
```

Press Ok, save the part and test the rule from the iLogic Parameters.

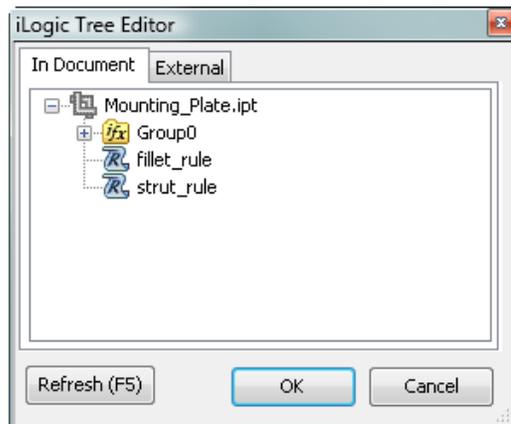
When strut is set to “2” the overall base, size needs to change as well. Edit the rule and add some overall parameters.

Access existing iLogics rules from the Tree Editor or the browser.

From the browser, drop down Model and choose iLogic



The Tree Editor is accessed by pressing the Edit iLogic Tree button from the Manage tab.

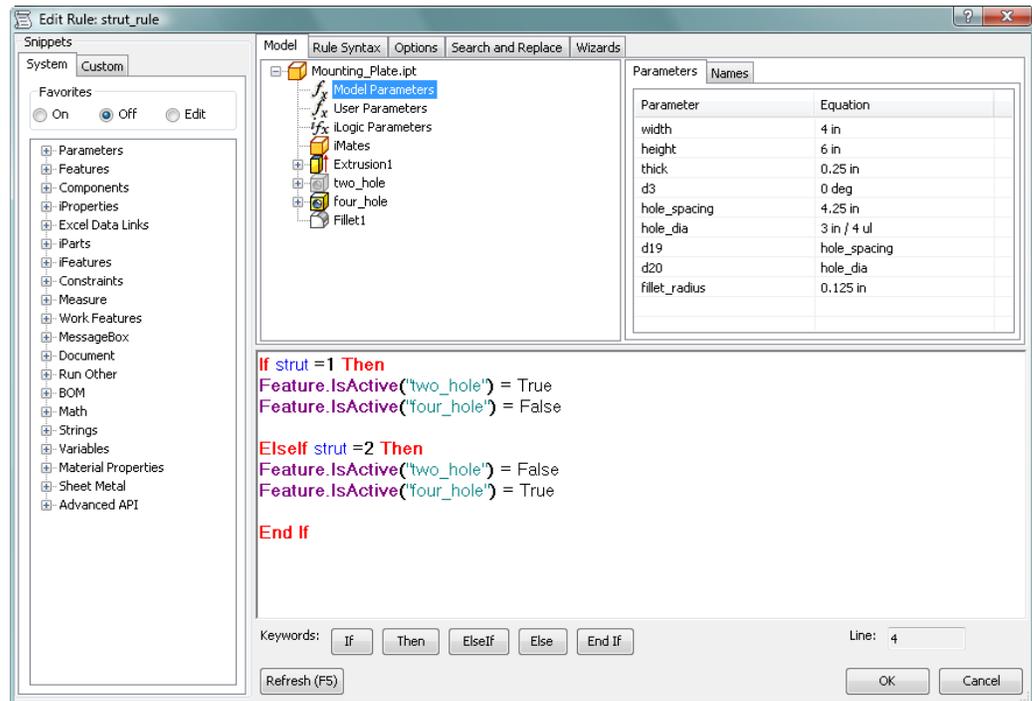


From either one, rules can be renamed and edited. Only the iLogic Tree Editor can change the order of rules.

Double click the **strut_rule** and the rule opens into the Rule Editor dialog.

Add some space and choose Model Parameters and from the Parameters tab width and height are visible. After making some room double clicking width will insert it into the rule window. Finish the line by adding = 4.

Copy and paste the width line to the second If statement. Change the 4 to a 6. The completed code should look like this.



```

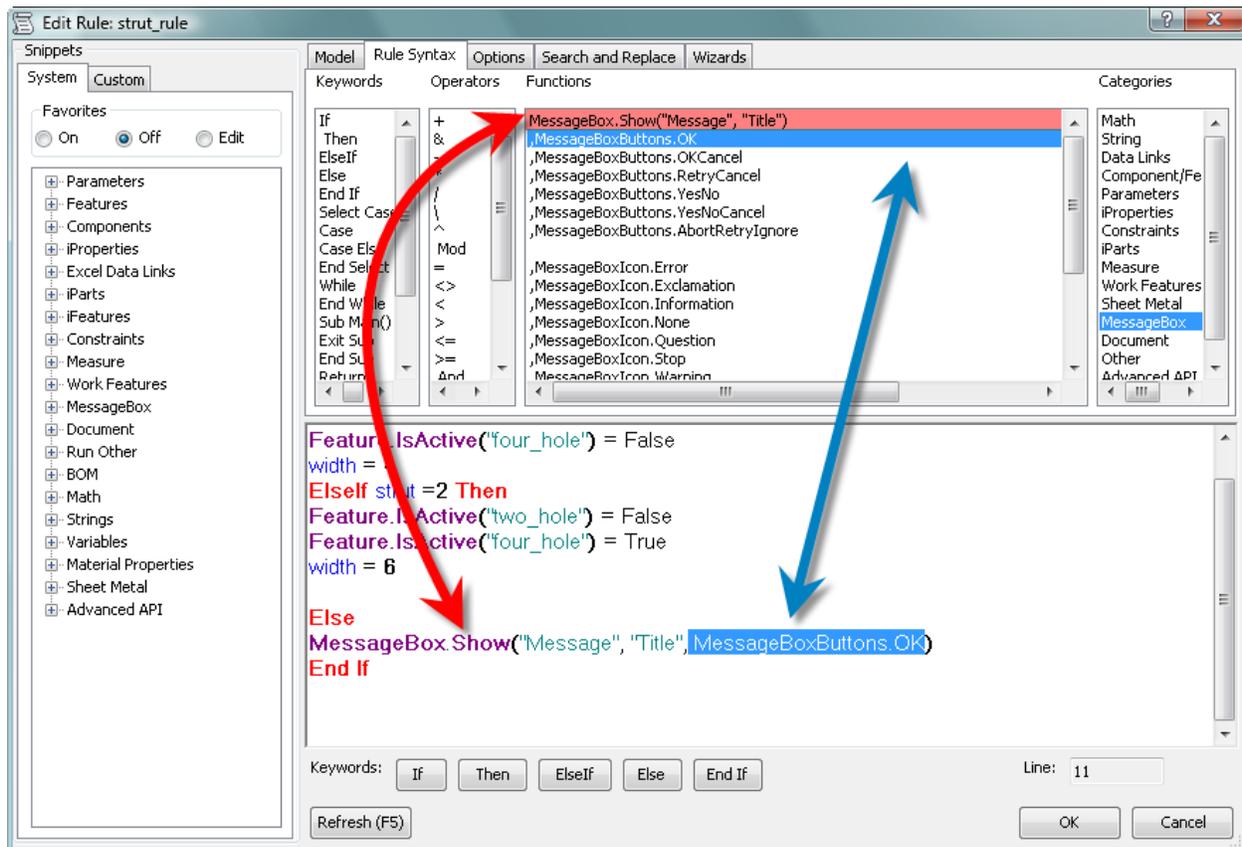
If strut =1 Then
Feature.IsActive("two_hole") = True
Feature.IsActive("four_hole") = False
width = 4
Elseif strut =2 Then
Feature.IsActive("two_hole") = False
Feature.IsActive("four_hole") = True
width = 6
End If
    
```

ERROR DIALOG CREATION

Lets add an Else statement that produces an error dialog box if anything other than a 1 or 2 is entered.

Add a line between width=6 and End If. Press the **Else** button. Go to the Rule Syntax tab and change the Categories field to MessageBox. Double click the first line "**MessageBox.Show("Message", "Title")**" then move the cursor back to a few spaces till it is behind the last comma. Double click the 2nd line "**MessageBoxButtons, OK**" and the line will be added to the rules windows below.

See the image on the following page.



Change the Title to “ERROR” and the Message to ”Parameter strut can only equal 1 or 2. sturt has been changed back to 1”

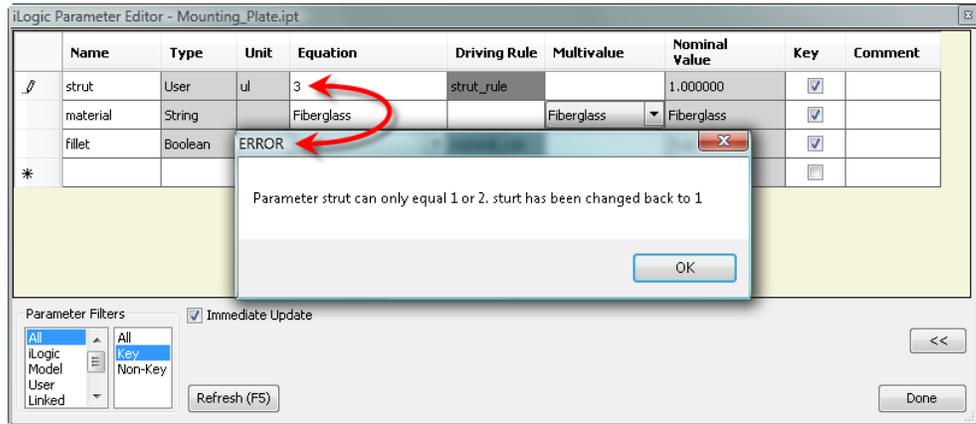
Lastly add a new line to change the parameter strut back to 1. It will simply be “strut=1”. Then copy the three lines from the first If statement.

The finished code should look like this.

```

If strut =1 Then
Feature.IsActive("two_hole") = True
Feature.IsActive("four_hole") = False
width = 4
ElseIf strut =2 Then
Feature.IsActive("two_hole") = False
Feature.IsActive("four_hole") = True
width = 6
Else
MessageBox.Show("Parameter strut can only equal 1 or 2. sturt has been changed back to 1",
"ERROR", MessageBoxButtons.OK)
strut=1
Feature.IsActive("two_hole") = True
Feature.IsActive("four_hole") = False
width = 4
End If
    
```

Press ok and update the model if needed. Open up the iLogic parameters and test the rule again. Change the value to 3 and the new dialog should open. Upon pressing OK button the value is change back to 1 and the part updates.



RULE #3 – MATERIAL AND STOCK NUMBER CHANGE

Add another rule by pressing the Add Rule button. Name it **material_rule**. This time use the material parameter. The first line should look like this.

If material = "Zinc-Plated Steel" Then

Next switch to the Rule Syntax and change the Category to iProperties and add the following lines
iProperties.Material
iProperties.Value("Project", "Part Number")

Change Part Number to **Stock Number**. Not all iProperty fields are listed but any can be used. Change the two lines to match the following.

If material = "Zinc-Plated Steel" Then
iProperties.Material = "Zinc-Plated Steel"
iProperties.Value("Project", "Stock Number") = "Zinc-123"

The iProperties Material field will be updated to match followed by entering a custom Stock Number into the iProperties Stock Number field. Now press **Elseif** and copy and paste the first three lines. Change the material to Green-Painted Steel as shown below.

If material = "Zinc-Plated Steel" Then
iProperties.Material = "Zinc-Plated Steel"
iProperties.Value("Project", "Stock Number") = "Zinc-123"
Elseif material = "Green-Painted Steel" Then
iProperties.Material = "Green-Painted Steel"
iProperties.Value("Project", "Stock Number") = "Green-123"

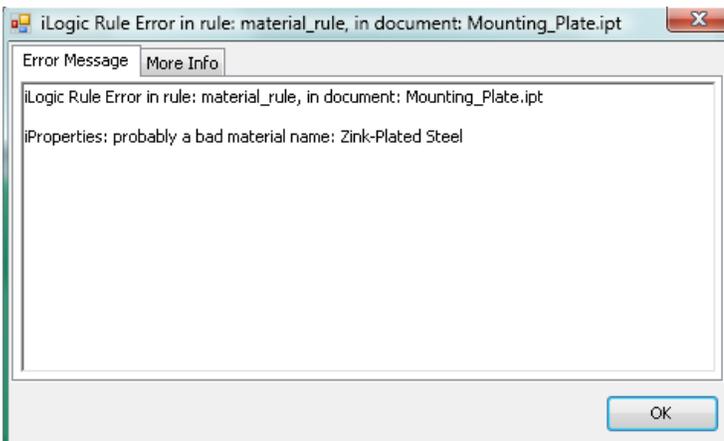
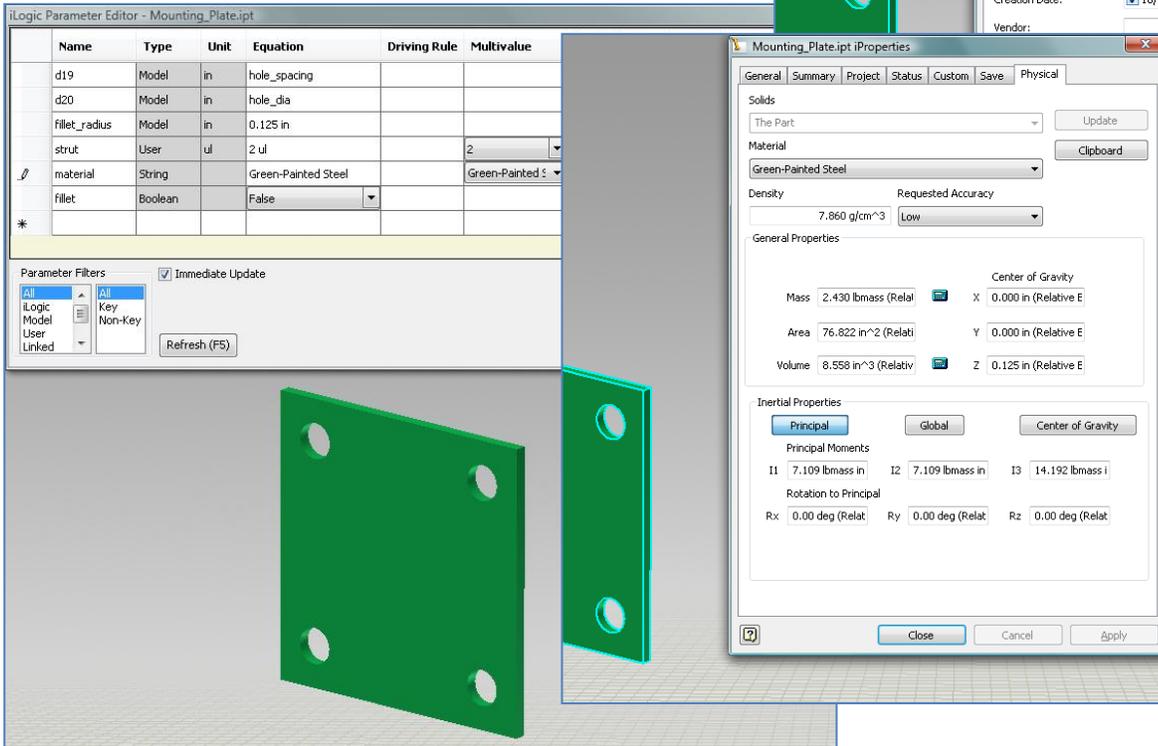
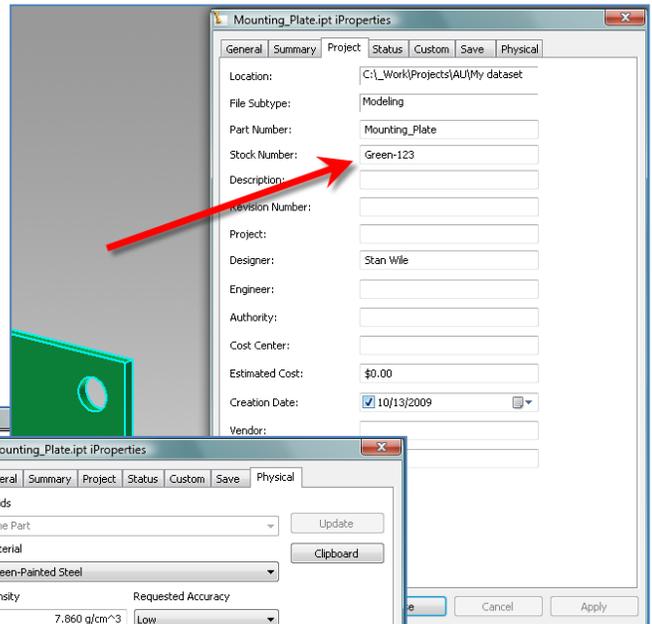
Repeat for the following materials, Galvanized Steel, 316 Stainless Steel, and Fiberglass. The final code should look like this.

If material = "Zinc-Plated Steel" Then
iProperties.Material = "Zinc-Plated Steel"
iProperties.Value("Project", "Stock Number") = "Zinc-123"
Elseif material = "Green-Painted Steel" Then
iProperties.Material = "Green-Painted Steel"
iProperties.Value("Project", "Stock Number") = "Green-123"
Elseif material = "Galvanized Steel" Then
iProperties.Material = "Galvanized Steel"

```

iProperties.Value("Project", "Stock Number") = "Galv-123"
Elseif material = "316 Stainless Steel" Then
iProperties.Material = "316 Stainless Steel"
iProperties.Value("Project", "Stock Number") = "SS-123"
Elseif material = "Fiberglass" Then
iProperties.Material = "Fiberglass"
iProperties.Value("Project", "Stock Number") = "Fiber-123"
End If
    
```

Test the rule from the iLogic Properties again. Here I am showing Green-Painted Steel.



It should be noted that all Material types used here were created before hand in this Part. If a material is entered that doesn't exist, when the part is changed to that material the error shown to the left will appear.

Rules can reference other rules. Edit the **material_rule** again and add to each If statement the iLogic parameter "**fillet**." It will be followed by = **True** on all materials but fiberglass. Grouping rules like this allows more control and makes for far less typing across rules. Only the fillet parameter needs to be changed from True to False and any number of things can take place under the fillet rule. We will later add to the Fillet rule.

So the Rule should look like this by now.

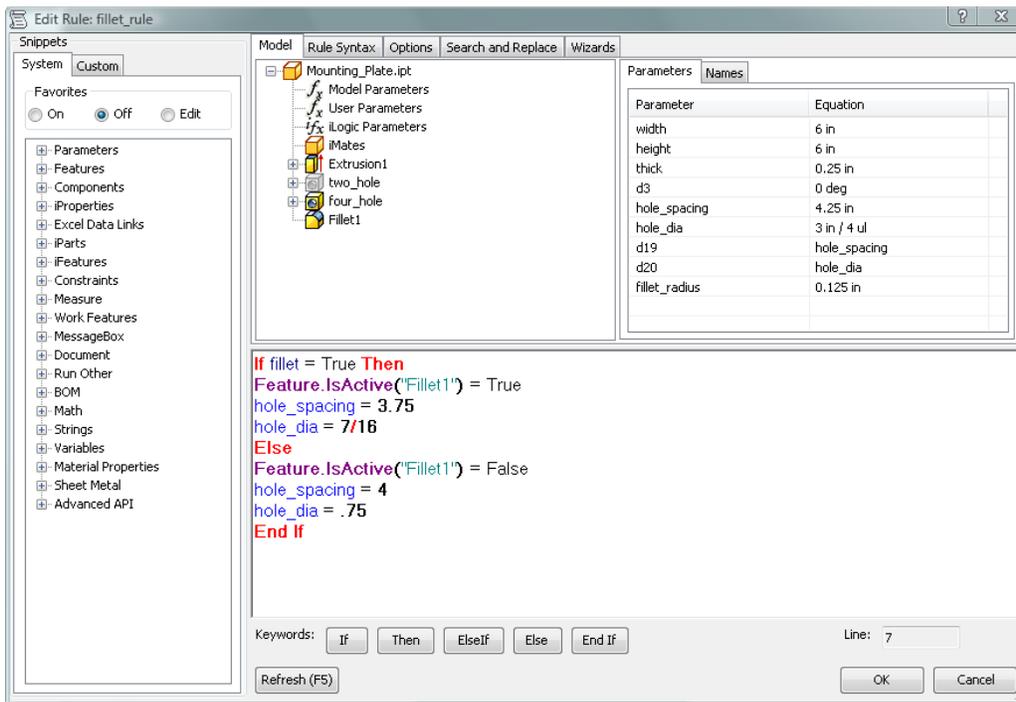
```

If material = "Zinc-Plated Steel" Then
iProperties.Material = "Zinc-Plated Steel"
iProperties.Value("Project", "Stock Number") = "Zinc-123"
fillet = True
Elseif material = "Green-Painted Steel" Then
iProperties.Material = "Green-Painted Steel"
iProperties.Value("Project", "Stock Number") = "Green-123"
fillet = True
Elseif material = "Galvanized Steel" Then
iProperties.Material = "Galvanized Steel"
iProperties.Value("Project", "Stock Number") = "Galv-123"
fillet = True
Elseif material = "316 Stainless Steel" Then
iProperties.Material = "316 Stainless Steel"
iProperties.Value("Project", "Stock Number") = "SS-123"
fillet = True
Elseif material = "Fiberglass" Then
iProperties.Material = "Fiberglass"
iProperties.Value("Project", "Stock Number") = "Fiber-123"
fillet = False
End If

```

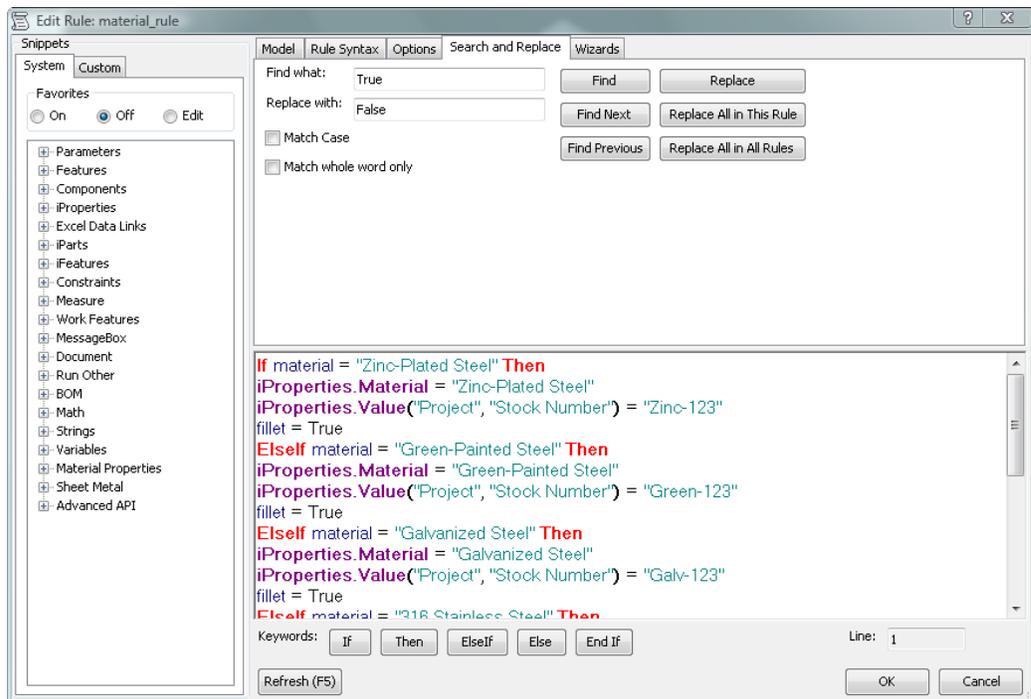
Test this to make sure the fillet feature is suppressed when set to fiberglass and no other materials.

On our sample part the fiberglass part has smaller holes and a different hole spacing. To make this change, edit the **fillet_rule**. Pull in the Model Parameters named **hole_spacing**, and **hole_dia** as on the next page.



Press OK and save the part. Then test the part by changing the material iLogic parameter. After testing you will see the small holes and fillets are on everything but the Fiberglass. **Exactly the opposite of what was required.** This would be a lot of typing had we not used the fillet parameter inside of the material_rule. To fix this edit the **material_rule**. Every fillet that = True needs to be changed to False. Instead of typing over every line use the Search and Replace tab.

Type True in the Find what box and False in the Replace with box and press Replace All in This Rule. Remember to change the last line from False to True. Press OK and test the rule again. This time Fiberglass is the only one with small holes and fillets.



Hopefully that gets you well on your way with using iLogic.

The completed bracket assembly instructions will be posted in pdf form and available for download from the AU website. The assembly instructions document will be reviewed along with the completed assembly.iam if class time is available.