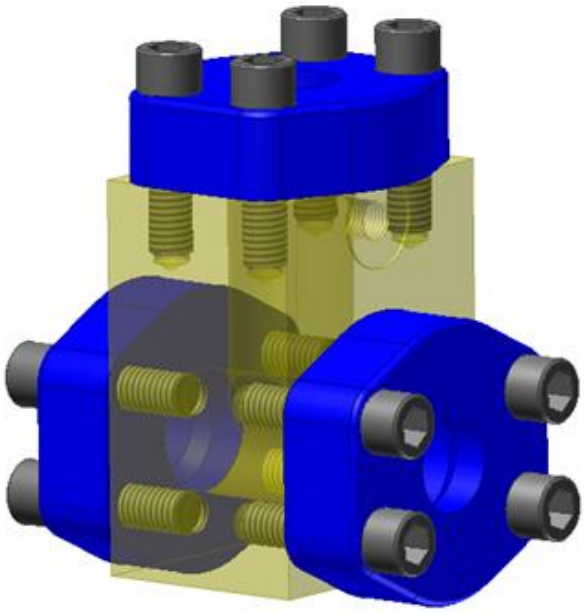
INTRODUCTION

In this tutorial, you will build an assembly using Autodesk® Inventor® iLogic.



The skills you will learn are:

- How to pass information from an assembly to its components
- How to drive iPart configurations
- How to change component pattern dimensions
- How to suppress/unsuppress components
- How to suppress/unsuppress constraints
- How to write data to an Excel spreadsheet
- How to update iProperties

This tutorial assumes the knowledge of basic Inventor assembly modeling techniques such as constraint creation. If you are not familiar with these concepts, please take the time to familiarize yourself with them before proceeding with this tutorial.

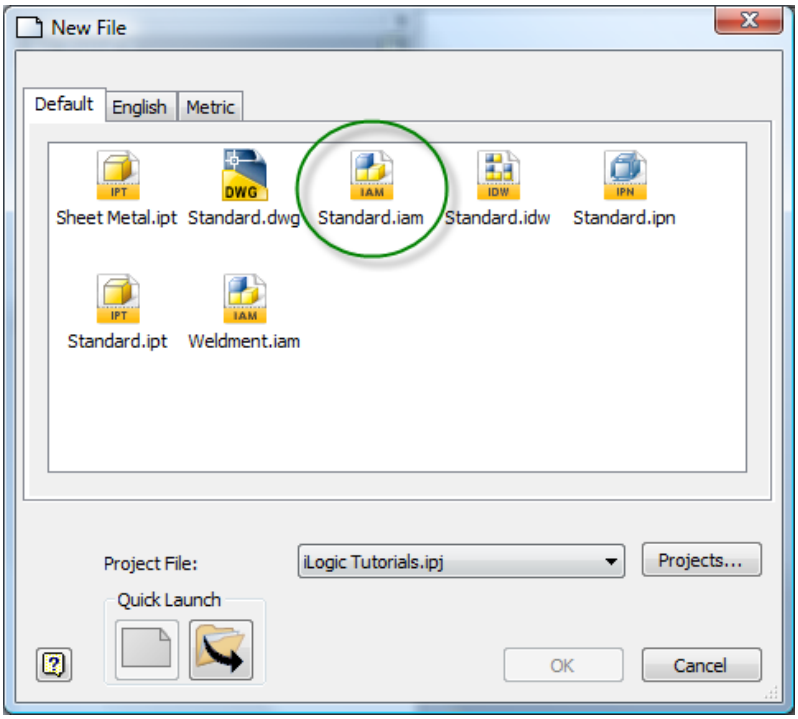
This tutorial requires completion of Tutorial 2 (Manifold Block Part).

Autodesk®

START A NEW ASSEMBLY FILE

Close any open Inventor files, and make sure the "iLogic Tutorials 2009" project is active. This should have been installed along with Inventor iLogic.

Open a new Inventor assembly using the **Standard.iam** template.



ASSEMBLY PROCESS OVERVIEW

Let's examine the steps you take to create the assembly:

- Determine how the design will be used in the present application, and how it could be used in future applications.
- Write the rules that you desire to have in your assembly in ordinary language (i.e., Plain English). These plain language rules we serve as a guide as you create the actual rules in the design using iLogic Rule language.

After you have written the rules in plain language, place the components into the assembly and:

- Constrain the components and examine the remaining degrees of freedom.
- Add more constraints as necessary to fully constrain all parts.

Autodesk®

PLACE THE FIRST COMPONENT

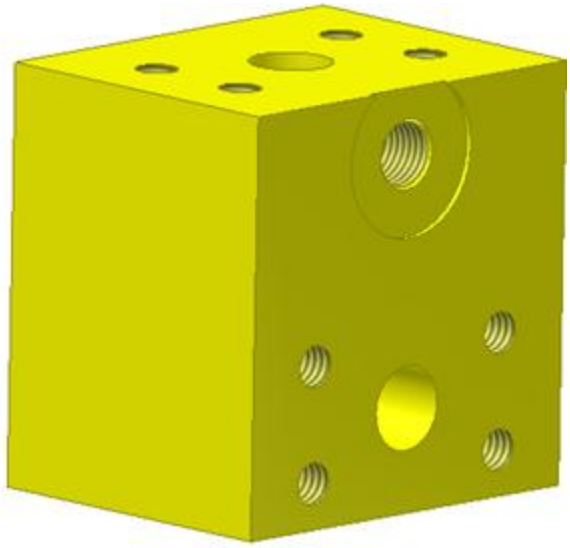
In this tutorial the assembly is a simple manifold block with flange fittings that are held on with socket head cap screws. The block is the first part placed into the assembly.

Click the **Place Component** tool from the Inventor Assembly Panel.

- In the Open dialog box, double-click **manifold_block.ipt**.
  (This is the part file you created in Tutorial 2.)

A grounded occurrence of the component is placed in the assembly with the part origin aligned with the assembly origin. A second occurrence of the part is attached to your cursor in the event that you want to place another. In this case you do not.

- Right-click the graphics window and select **Done**.



ADD OTHER COMPONENTS

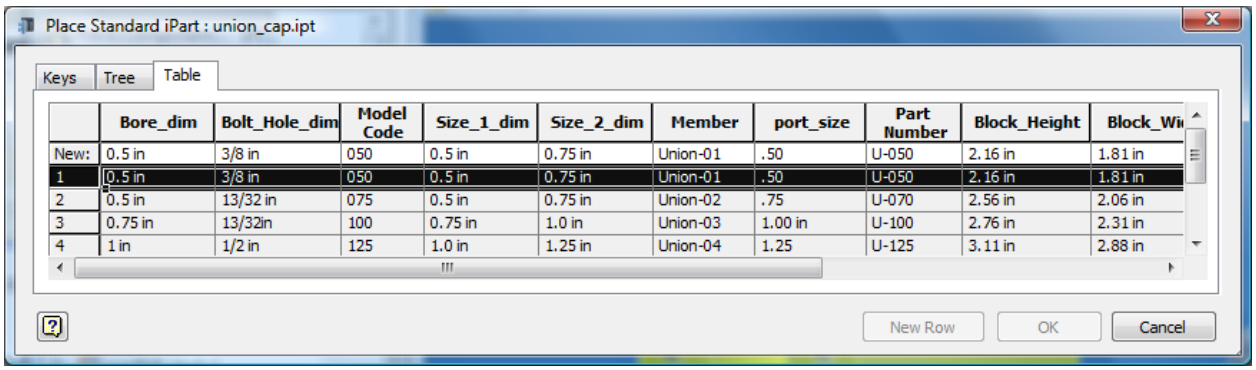
In addition to the block, this assembly will consist of a set of 3 union caps, and 3 sets of screws to attach the caps.

ADD UNION_CAP COMPONENTS

Click the Place Component tool. 

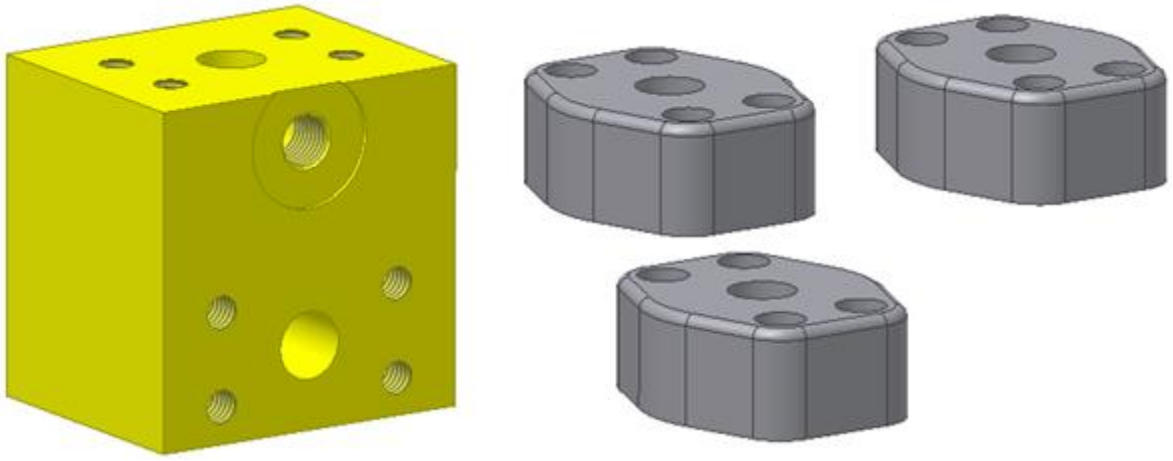
Double-click **union_cap.ipt**.

In the Place Standard iPart dialog box, choose the iPart member **Union-01** (Part Number U-050) under the Table tab.



	Bore_dim	Bolt_Hole_dim	Model Code	Size_1_dim	Size_2_dim	Member	port_size	Part Number	Block_Height	Block_Wi
New:	0.5 in	3/8 in	050	0.5 in	0.75 in	Union-01	.50	U-050	2.16 in	1.81 in
1	0.5 in	3/8 in	050	0.5 in	0.75 in	Union-01	.50	U-050	2.16 in	1.81 in
2	0.5 in	13/32 in	075	0.5 in	0.75 in	Union-02	.75	U-070	2.56 in	2.06 in
3	0.75 in	13/32in	100	0.75 in	1.0 in	Union-03	1.00 in	U-100	2.76 in	2.31 in
4	1 in	1/2 in	125	1.0 in	1.25 in	Union-04	1.25	U-125	3.11 in	2.88 in

Click in the graphics window to place the union near the manifold block. Add three unions, then Dismiss the Place Standard iPart dialog.

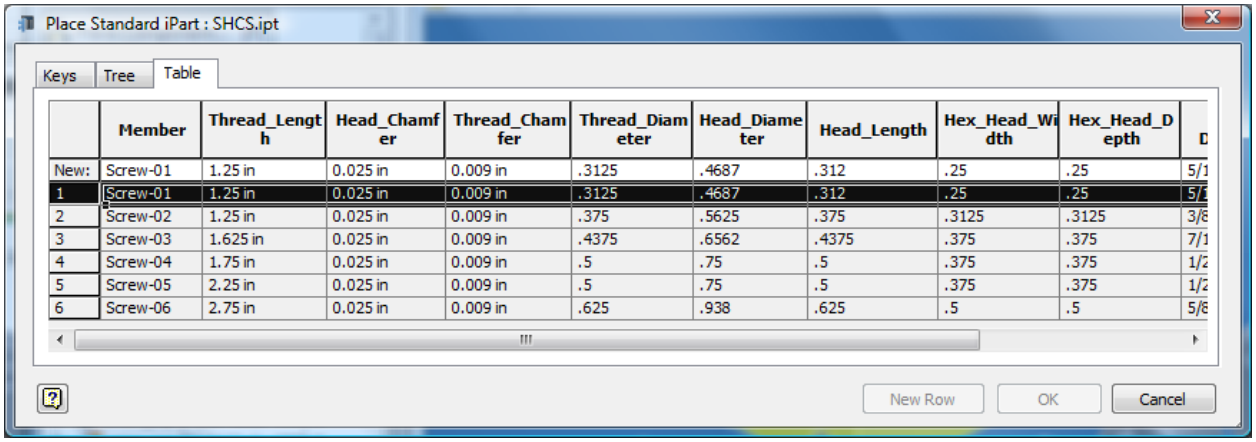


ADD SCREW COMPONENTS

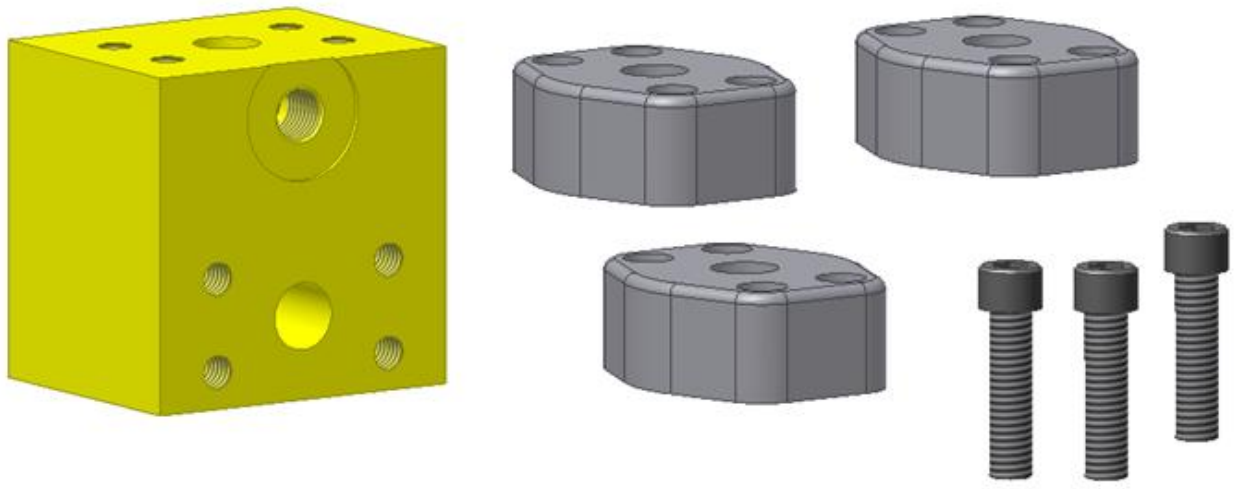
Click the Place Component tool again.

Double-click **Screw.ipt**.

In the Place Standard iPart dialog box, identify the iPart member **Screw-01**. Place three Screws in the graphics window, and then Dismiss the Place Standard iPart dialog.



	Member	Thread_Length	Head_Chamfer	Thread_Chamfer	Thread_Diameter	Head_Diameter	Head_Length	Hex_Head_Width	Hex_Head_Depth	D
New:	Screw-01	1.25 in	0.025 in	0.009 in	.3125	.4687	.312	.25	.25	5/1
1	Screw-01	1.25 in	0.025 in	0.009 in	.3125	.4687	.312	.25	.25	5/1
2	Screw-02	1.25 in	0.025 in	0.009 in	.375	.5625	.375	.3125	.3125	3/8
3	Screw-03	1.625 in	0.025 in	0.009 in	.4375	.6562	.4375	.375	.375	7/1
4	Screw-04	1.75 in	0.025 in	0.009 in	.5	.75	.5	.375	.375	1/2
5	Screw-05	2.25 in	0.025 in	0.009 in	.5	.75	.5	.375	.375	1/2
6	Screw-06	2.75 in	0.025 in	0.009 in	.625	.938	.625	.5	.5	5/8



SAVE THE FILE

Before you continue, save the file to a new name, and then close the current file.

- Click File > Save Copy As.
- Name your file **my_manifold_block.iam**
- Click Save

Now you're ready to begin assembling the components and establishing their relationships.

ASSEMBLE THE COMPONENTS

First, we'll change the colors of the union_cap components to make them stand out more prominently in the assembly:

- Click on the first union_cap component.
- From the color list, located at the far right on the Standard toolbar, select Blue.
- Repeat this for the other 2 union_cap components.

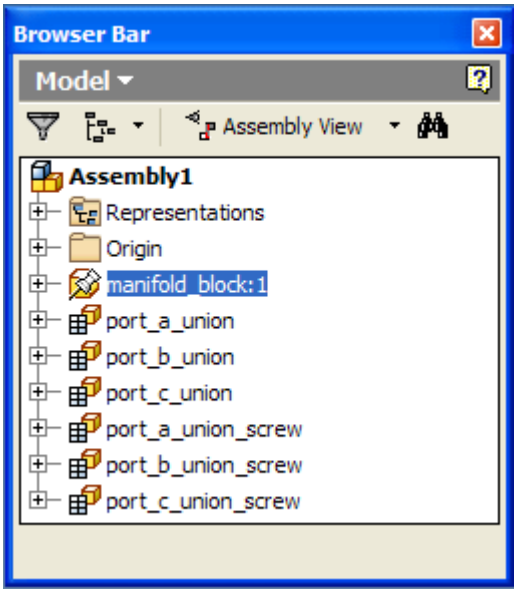
RENAME ASSEMBLY COMPONENTS

The components that have been added to the assembly so far are listed in the model browser. Presently the components' names are the same as the file names that they reference. We are going to change the component names in our assembly so that they better reflect their purpose, and to remove the iPart member identification.

The reason we need to do this is because if an iLogic rule is referencing a component

name, and that component name changes due to an iPart member change, the component name will be out of date in the rule, causing the rule to fail to execute properly.

- Click on **union_cap [Model Code = 050]:1** and change its name to **port_a_union**.
- Click on **union_cap [Model Code = 050]:2** and change its name to **port_b_union**.
- Click on **union_cap [Model Code = 050]:3** and change its name to **port_c_union**.
- Click on **Screw-01:1** and change its name to **port_a_union_screw**.
- Click on **Screw-01:2** and change its name to **port_b_union_screw**.
- Click on **Screw-01:3** and change its name to **port_c_union_screw**.



EDITING ILOGIC PARTS FROM WITHIN AN ASSEMBLY

Double click on **manifold_block:1**. Open the iLogic Parameter Editor by clicking the Parameters button in the iLogic toolbar or panel bar. Make sure that the following parameter values are set as indicated in the following table.

Parameter	Value
block	Tee
component_type	Standard
port_a_size	0.50

SAVE THE FILE

Before you continue, save the assembly file.

ADD ASSEMBLY CONSTRAINTS

Assembly constraints restrict the movement of components relative to each other. Apply all constraints necessary to fully constrain each part so as to leave no degrees of freedom.

MATE PORT A UNION CAP TO THE BLOCK

Create a Mate Constraint, between:

- The axis running through the center of Port A of the manifold block.
- The axis through the center hole of **port_a_union**.

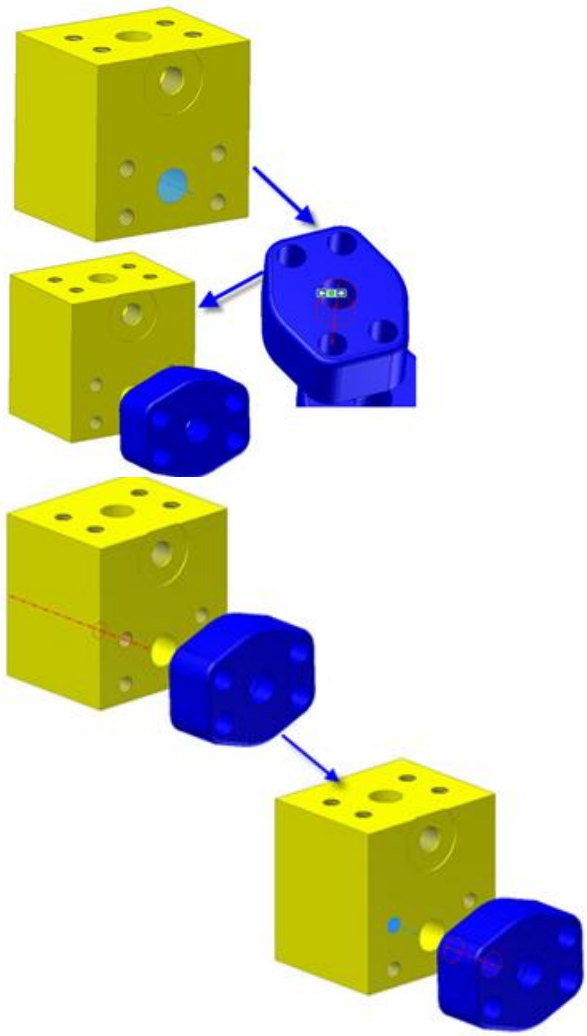
Note: If, after creating the constraint, the union cap is hidden within the block, click and drag the union cap away from the block. You can click and drag any part within an assembly and it moves if it is under-constrained.

Create another Mate Constraint, between:

- The axis running through the center of Port A top left screw hole on the manifold block.
- The axis along the center of the top left screw hole of **port_a_union**.

Create one more Mate Constraint, between:

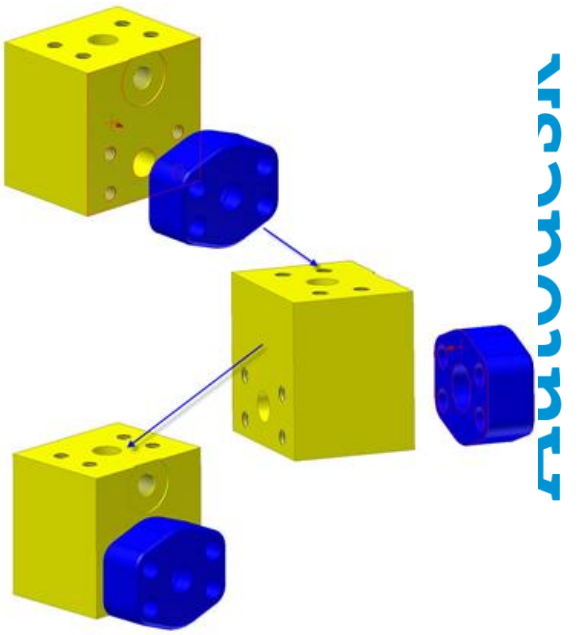
- The Port A face of the manifold block.
- The back face of **port_a_union**.



It's a good idea to give these constraints meaningful names, to make them easier to identify later on.

Name the three mate constraints we just created as follows:

- port_a_cap_center
- port_a_cap_hole
- port_a_cap_face



MATE PORT A SCREW TO THE UNION CAP

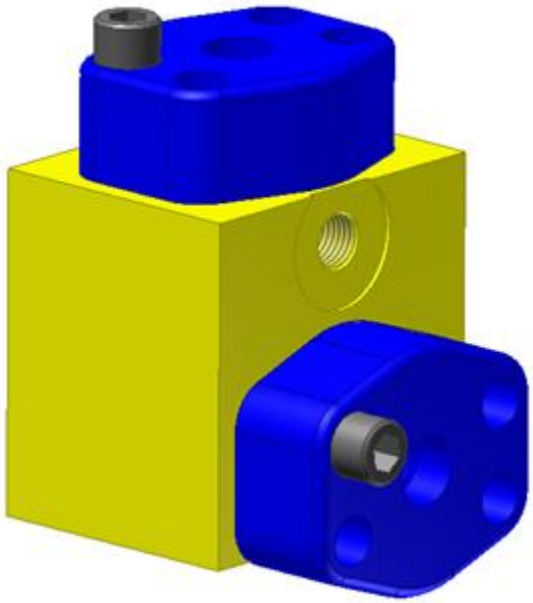
Create an Insert constraint, inserting:

- The **port_a_union_screw** into
- The top left screw hole in **port_a_union**.

Name this constraint "port_a_cap_screw".

PORT B & C UNION CONSTRAINTS

Follow the same steps as above for the Port B & C unions, and the Port B & C union screws. Use similar names for the constraints that you create. See image below:



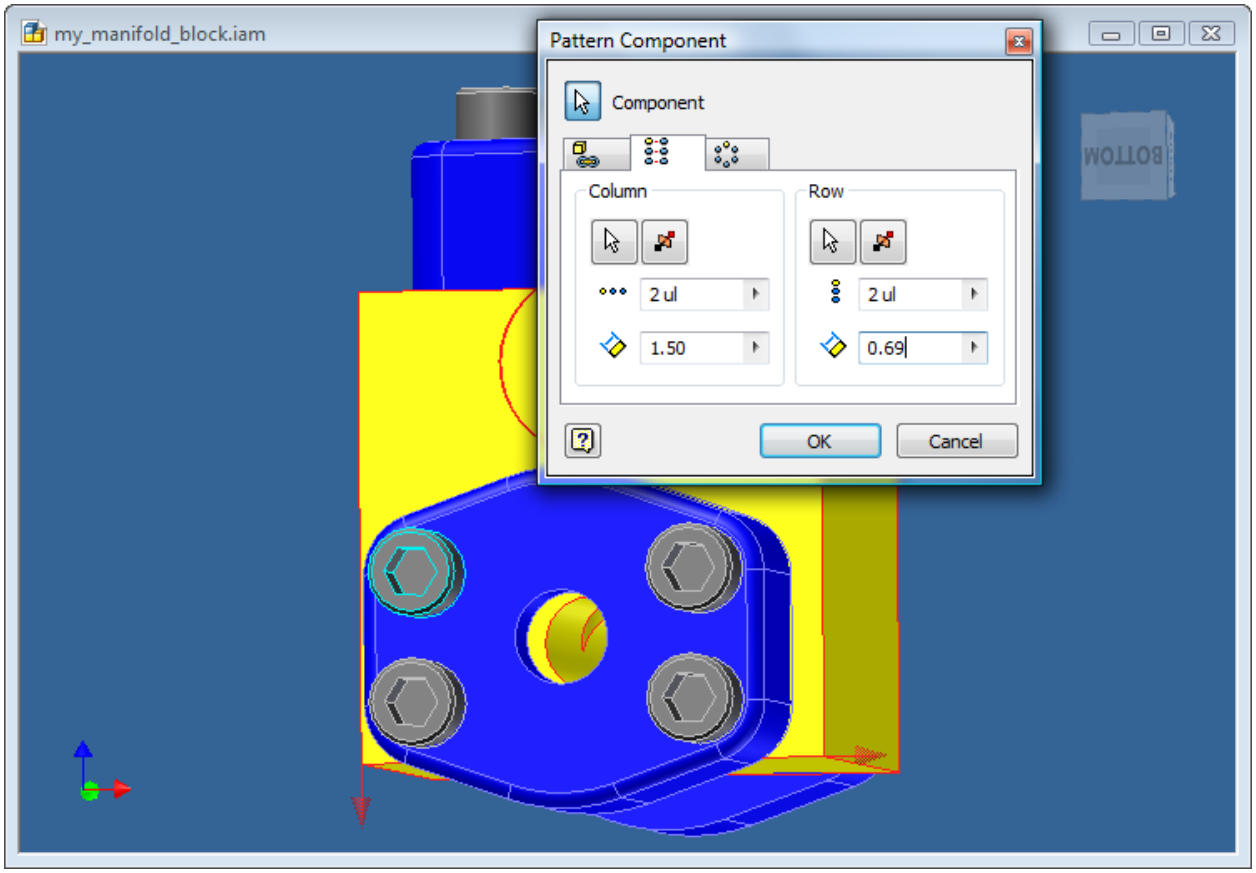
CREATE SCREW PATTERNS

For each of the ports on the manifold block, we will use a pattern of four screws to attach the union cap to the block. We'll use the corresponding screw that we've inserted into the assembly as the patterned component.

First, we'll create the screw pattern for Port A.

- Click the Pattern Component button on the Assembly panel bar to create a component pattern of the screw
- Click on **port_a_union_screw**

- Click the Rectangular tab
- Click on the Column Direction Arrow
- Click on the bottom horizontal edge of the Port A face
- Enter 1.50 for the Horizontal distance
- Click on the Row Direction arrow
- Click on the left vertical side of the Port A face
- Enter .69 for the Vertical distance



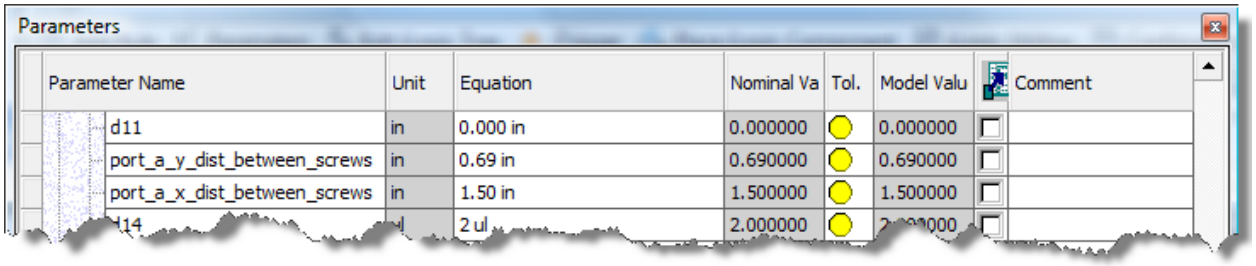
- Click OK in the Pattern Component Dialog box.
- Rename Component Pattern 1 in the Model Browser to be **port_a_screw_pattern**.

ASSIGN DESCRIPTIVE NAMES TO THE PATTERN-CONTROLLING PARAMETERS

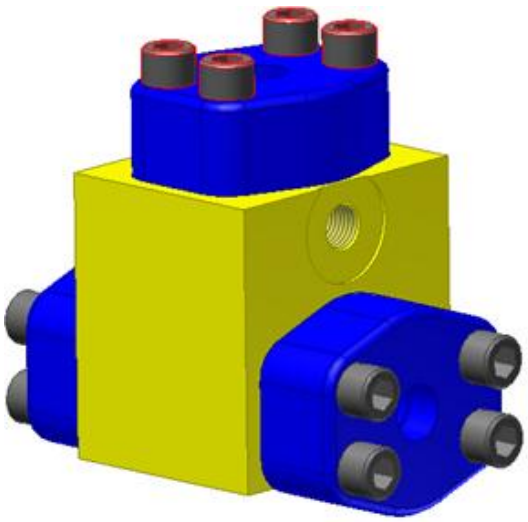
Creating the screw component pattern resulted in new model parameters being created. We need to rename these parameters with meaningful names for future use in this tutorial. Using the Inventor or iLogic Parameter Editor:

- Click on the parameter name with 0.69 in in the equation
- Change the parameter name to **port_a_y_dist_between_screws**

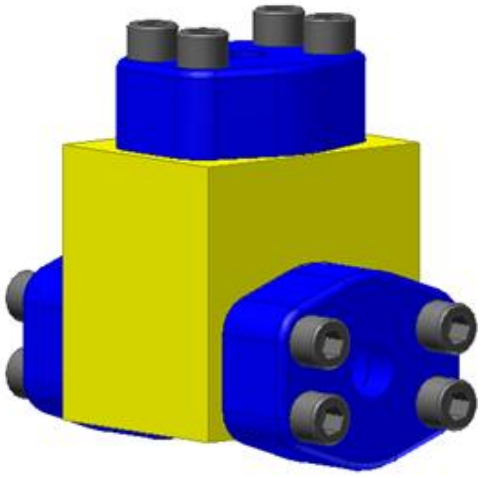
- Click on the parameter name with 1.50 in in the equation
- Change the parameter name to **port_a_x_dist_between_screws**



Repeat the above for Port B union screw and Port C union screws. Remember to rename the pattern when you are done. Your model should look like this when you are done:



Front View



Back View

SAVE THE FILE

Before you continue, save the file.

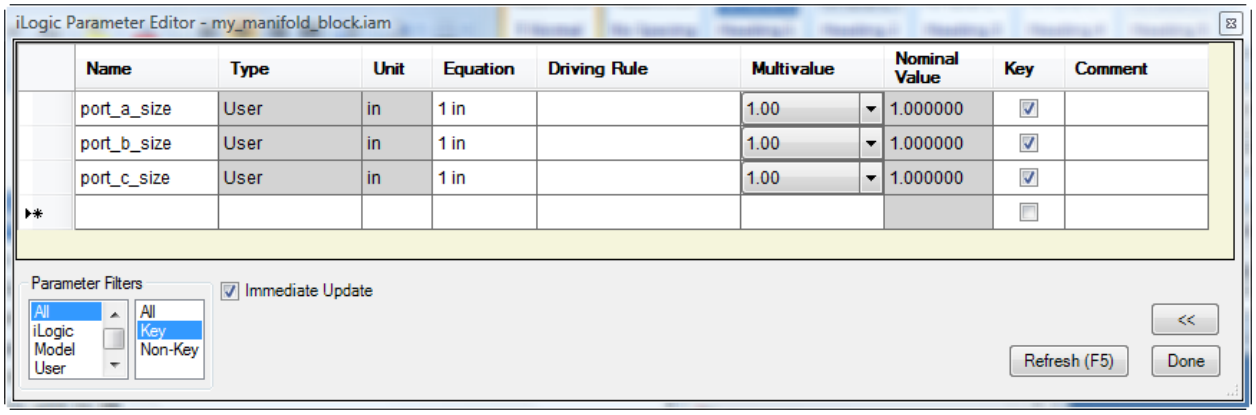
ADD CONTROL PARAMETERS FOR ASSEMBLY

Since this is a new assembly document, we need to add some parameters using the iLogic Parameter Editor. For the following steps, open the iLogic Parameter Editor.

CREATE PARAMETERS TO CONTROL PORT SIZES

- Create a new parameter named “port_a_size”
- Set its type to **User**
- Set its unit to **in**
- Make it a Multi-value list, with the values:
  0.50, 0.75, 1.00, 1.25, 1.50, 2.00, 2.50, 3.00
- Make it a key parameter

Repeat the above to create 2 more parameters: **port_b_size** and **port_c_size**.



CREATE PARAMETERS TO CONTROL COMPONENT AND BLOCK TYPE

Create a parameter called **component_type**

- Set the type to String
- Create a Multi-Value list of: standard, custom
- Set the Multi-Value list to standard
- Make this a Key Parameter

Create a parameter called **block**

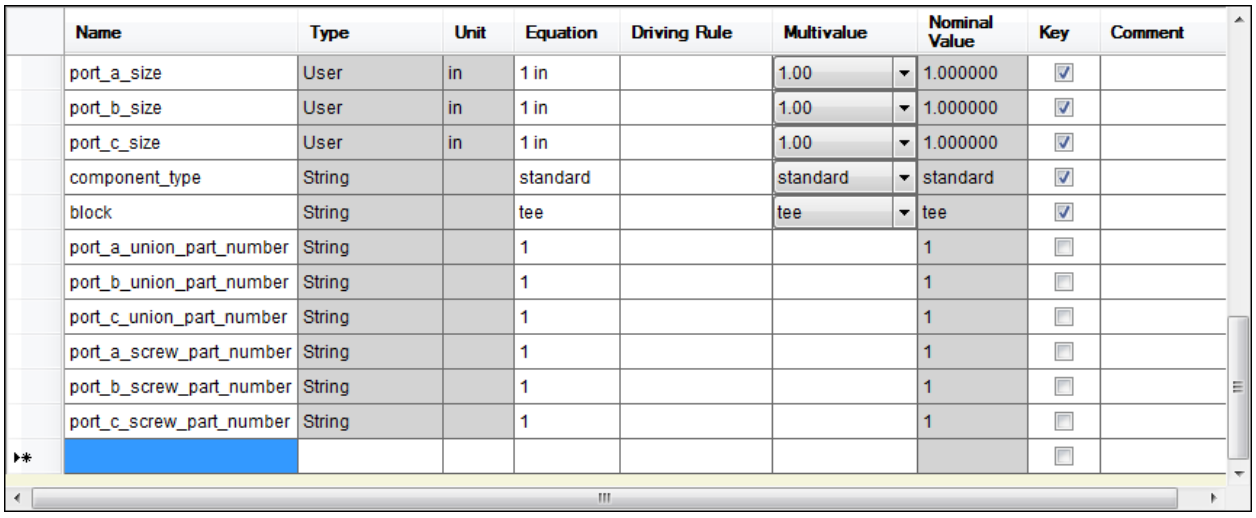
- Set the type to String
- Create a Multi-Value list of: tee, elbow
- Set the Multi-Value list to tee
- Make this a Key Parameter

CREATE PARAMETERS TO CONTROL COMPONENT PART NUMBERS

Create 7 additional parameters. For each, set the type to String, and the value to 1. Call these:

- port_a_union_part_number
- port_b_union_part_number
- port_c_union_part_number
- port_a_screw_part_number
- port_b_screw_part_number
- port_c_screw_part_number

Your entire set of new parameters should look like this:



Name	Type	Unit	Equation	Driving Rule	Multivalue	Nominal Value	Key	Comment
port_a_size	User	in	1 in		1.00	1.000000	☑	
port_b_size	User	in	1 in		1.00	1.000000	☑	
port_c_size	User	in	1 in		1.00	1.000000	☑	
component_type	String		standard		standard	standard	☑	
block	String		tee		tee	tee	☑	
port_a_union_part_number	String		1			1	☐	
port_b_union_part_number	String		1			1	☐	
port_c_union_part_number	String		1			1	☐	
port_a_screw_part_number	String		1			1	☐	
port_b_screw_part_number	String		1			1	☐	
port_c_screw_part_number	String		1			1	☐	
							☐	

SAVE THE FILE

Before you continue, save the file.

CREATE RULES IN THE ASSEMBLY

The following sections create a series of rules to manage the contents of the assembly. In these sections, we will provide excerpts of the rule text, which you can copy and paste into the iLogic rule editor. Keep in mind that it is always possible to utilize the helpers in the various tabs of the rule editor to assist you as you enter the rule text. We'll provide notes to guide you to important areas as they're introduced. It's up to you whether you want to use the copy-and-paste approach, or whether you want to enter the rule text by hand.

The appendix at the end of this document provides the complete text of all of the rules.

PASS PARAMETERS FROM THE ASSEMBLY TO THE PARTS.

This model has a part with iLogic rules in it: **manifold_block:1**. We will need to pass the assembly level parameter to the part. To do this:

- Create a new rule named "assembly_to_parts_rule"

This rule sets corresponding parameters in the part based on the values of the control parameters in the assembly. Notice how use use the form of the Parameter function that specifies the component name, as well as the parameter name.

```
Parameter("manifold_block:1", "block") = block
Parameter("manifold_block:1", "component_type") = component_type
Parameter("manifold_block:1", "port_a_size") = port_a_size
Parameter("manifold_block:1", "port_b_size") = port_b_size
Parameter("manifold_block:1", "port_c_size") = port_c_size
```

Press OK when you have completed this rule.

EDITING PART LEVEL RULES IN AN ASSEMBLY

In the Manifold Block Part tutorial we added a rule to the manifold block part to control changing from tee to elbow. This rule is great when we are just using it in the manifold block by itself, but we will need it at the assembly level as well. Instead of rewriting a rule that already exists, let's copy the original.

- Double click on manifold_block:1 from the model browser. (the other

components should become transparent)

- Click on the iLogic Tree Editor icon.
- Double click on **component_type_rule**.
- Highlight all of the rule text and copy it.
- Cancel out of the rule editor.
- Cancel out of the iLogic Tree Editor.
- Double click on my_manifold_block.iam in the model browser
- Add a new rule named “component_type_rule”.
- Paste the rule text that we copied from the block’s component_type_rule into the rule editor. Your rule text should look like this:

```
If component_type = "standard" Then
port_b_size = port_a_size
port_c_size = port_a_size
End If
```

Now, click OK to save this assembly-level rule.

PORT A RULE

There will be quite a bit going on with Port A when we change the port size. We will need to change the port size, update its iPart number, change the screw size if required, screw location, and screw kit part number.

Let’s add a rule to do this.

- Make sure the manifold block assembly is active, and click the Add Rule icon.
- Name this rule “port_a_rule”.

The first part of this rule will adjust the screw pattern spacing, based on information stored in the union part’s iPart table. We can look up the row in use based on the **port_a_size** parameter, and then assign values to two different assembly parameters from two other columns.

```
i = iPart.FindRow("port_a_union", "port_size", "=", port_a_size)
port_a_y_dist_between_screws = iPart.CurrentRowValue("y_dist_betwn_screw")
port_a_x_dist_between_screws = iPart.CurrentRowValue("x_dist_betwn_screw")
```

The iPart-related statements used above can be found on the iParts category under the Rule Syntax tab.

The next part of the rule will select the appropriate iPart row inside the screw part

based on the selected port size. In this section, we use a series of “if” statements to set the appropriate iPart member (“Screw-01”, etc.), according to the current value of the “port_a_size” parameter.

```
If port_a_size = .50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-01")
ElseIf port_a_size = 0.75 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-02")
ElseIf port_a_size = 1.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-02")
ElseIf port_a_size = 1.25 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-03")
ElseIf port_a_size = 1.50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-04")
ElseIf port_a_size = 2.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-04")
ElseIf port_a_size = 2.50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-05")
ElseIf port_a_size = 3.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-06")
End If
```

The last part of this rule for Port A will get the part number for the flare flange and store it in an assembly parameter, where we will use it in another rule later on.

```
port_a_union_part_number = iProperties.Value("port_a_union", "Project", "Part Number")
```

SAVE THE FILE

Before you continue, save the file.

PORT B RULE

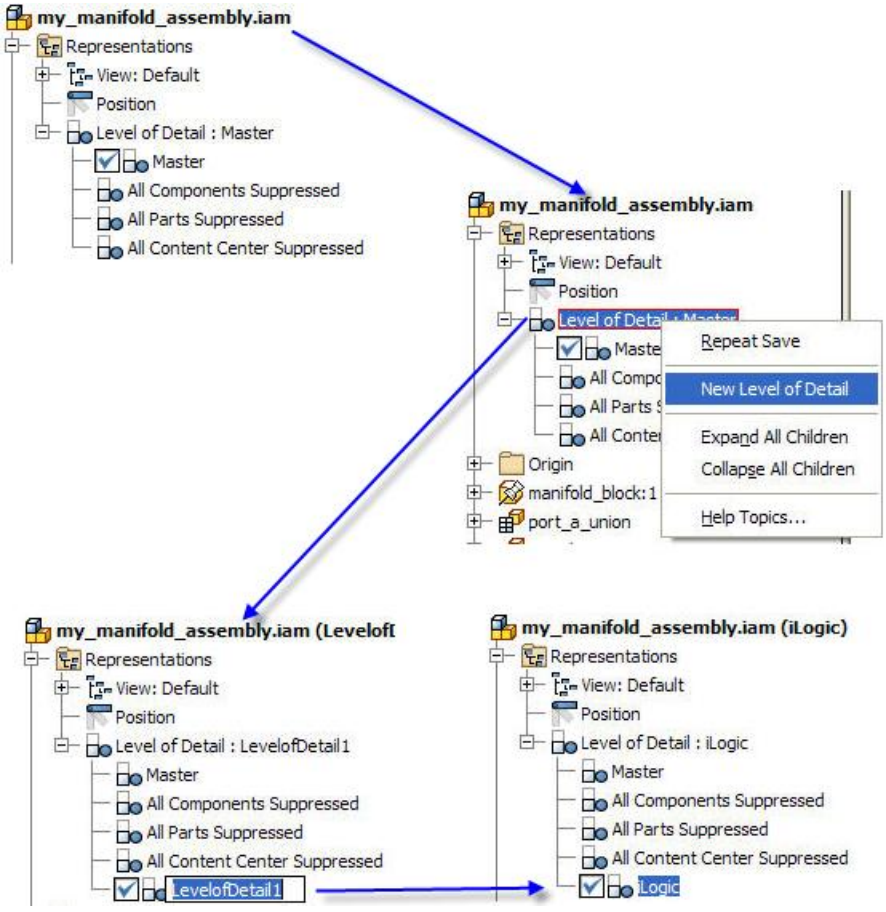
Port B is a special case compared to ports A and C. This is because port B does not exist when the manifold block is an elbow style. If the manifold block is an elbow style, we will need to suppress the union cap and the union screws used for Port B. We will also need to suppress the mates associated to the union cap. One other item before we go on with writing a rule is that we have to set a level of detail because we will be suppressing components. Rules that affect level of detail related items in an assembly require a custom level of detail to be defined and saved before they can be written or iLogic will generate an error message to that effect.

SETTING A LEVEL OF DETAIL

- In the model browser, expand the Representations folder and the Level of Detail

folder.

- Right-click the Level of Detail node in the browser, and select new Level of Detail.
- A new level of detail will be added.
- Click the Level of Detail representation “LevelofDetail1” and rename it to “iLogic”.



PORT B RULE CONTINUED…

Let's begin to write this rule. Create a new rule named "port_b_rule". Populate the rule as follows.

The first thing that we'll do is decide whether we're making a tee style block, and remember that in a separate variable. We'll then use this variable later to set other parameters.

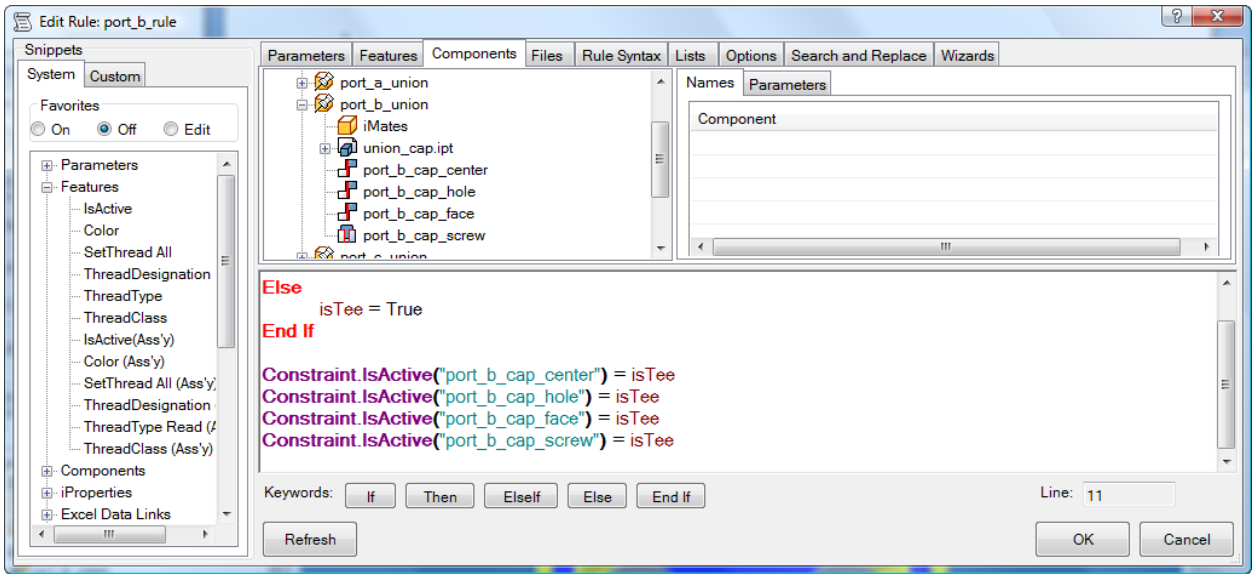
```
If block = "elbow" Then
        isTee = False
Else
        isTee = True
End If
```

We'll use the value of the **isTee** variable later – it holds a True or False value to indicate whether we are making a tee manifold.

We are now going to add lines to our rule that will turn off the constraints that locate the union and union screw when the manifold block is an elbow, and turn them on when we're making a tee style. Notice how we can use the **isTee** variable to turn these constraints on or off according to the value of the **block** parameter.

```
Constraint.IsActive("port_b_cap_center") = isTee
Constraint.IsActive("port_b_cap_hole") = isTee
Constraint.IsActive("port_b_cap_face") = isTee
Constraint.IsActive("port_b_cap_screw") = isTee
```

Notice how specifying the names of these constraints has made it easier to refer to them in this rule. Also keep in mind that you can use the model tree information provided on the Components tab of the rule editor to help fill in the names of the constraints.



Next, we add two additional lines to conditionally include the Port B Union part, and corresponding screw pattern:

```
Component.IsActive("port_b_union") = isTee
Component.IsActive("port_b_screw_pattern") = isTee
```

These both use the same "isTee" variable used previously. When the screw pattern is suppressed, the screw component itself is also suppressed.

Next, if we are using a Port B, we need to set the port size, the parameter values that control the screw pattern, and the Port B union part number.

```
if isTee Then
        i = iPart.FindRow("port_b_union", "port_size", "=", port_b_size)
        port_b_y_dist_between_screws = iPart.CurrentRowValue("y_dist_betwn_screw")
        port_b_x_dist_between_screws = iPart.CurrentRowValue("x_dist_betwn_screw")
        port_b_union_part_number = iProperties.Value("port_b_union", "Project", "Part
Number")
End If
```

Notice how we've enclosed this entire block in an "If isTee" statement, so that these lines are only processed if we're using a tee-style manifold block. The statement "if isTee then" is the equivalent of "if isTee = True Then" but provides a more concise expression format.

Here, we first choose the appropriate row in the union part's iPart table, corresponding to the value of the **port_b_size** parameter, and then extract the values that are to be used for the x and y pattern offsets. Then, we extract the Part Number from the union part, and store its value in another parameter for later reference.

CHOOSING THE PORT B SCREW SIZE

The last part of this rule will choose the member within the screw part's iPart table to use for Port B. This will be based on the value of the **port_b_size** parameter. We use a series of If/Then/Else statements to control this.

```
If port_b_size = .50 then
        iPart.ChangeRow("port_b_union_screw", "Screw-01")
elseif port_b_size = .75 then
        iPart.ChangeRow("port_b_union_screw", "Screw-02")
elseif port_b_size = 1.00 then
        iPart.ChangeRow("port_b_union_screw", "Screw-02")
elseif port_b_size = 1.25 then
        iPart.ChangeRow("port_b_union_screw", "Screw-03")
elseif port_b_size = 1.50 then
        iPart.ChangeRow("port_b_union_screw", "Screw-04")
elseif port_b_size = 2.00 then
        iPart.ChangeRow("port_b_union_screw", "Screw-04")
elseif port_b_size = 2.50 then
        iPart.ChangeRow("port_b_union_screw", "Screw-05")
elseif port_b_size = 3.00 then
        iPart.ChangeRow("port_b_union_screw", "Screw-06")
end if
```

We could have included all of these lines within the "If isTee" block above, since the screw only matters if we're actually including it in our model. To simplify the rule, however, we've kept these lines separate which has no effect on the results the rule produces.

This completes the **port_b_rule**. Press OK to close the dialog box and save the rule.

SAVE THE FILE

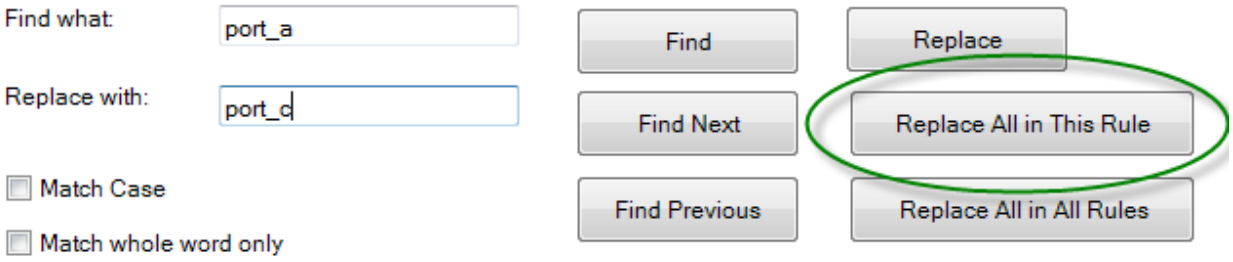
Before you continue, save the file.

PORT C RULE

The rule for Port C is almost exactly the same as the rule for Port A, with the difference being that everything that referenced Port A needs to reference Port C instead. The quick way to create this rule would be to copy and paste, find and replace the Port A references with Port C references. Let's get started.

- Click on the iLogic Tree Editor
- Double click on port_a_rule

- Using your left mouse button, highlight the entire rule
- Press Ctrl+C to copy the rule text
- Click OK on the rule editor dialog box
- Click OK to close the iLogic Tree Editor
- Click Add Rule
- Enter the name port_c_rule
- Click OK
- In the rule editor click in the rule text pane so your cursor appears
- Press Ctrl+V to paste the rule that we copied from port_a_rule
- Click on the Search and Replace tab



- In Find what: type **port_a**
- In Replace with: type **port_c**
- Check off Match Case
- Check off Match whole word only
- Click on **Replace All in This Rule**
- Click OK on the dialog box, this rule is up to date.

SAVE THE FILE

Before you continue, save the file.

CALCULATING PART NUMBERS

Two final pieces of information we need to determine for our model is the part numbers to use for the union screws and elbows. To do this, we'll need to create some additional rules in our assembly.

SCREW PART NUMBER RULE

The first part of this rule will calculate a part number to be used for the Port A screw, based on the value of the **port_a_size** parameter. In this rule, we introduce a new statement – Select Case – which can be used to execute one of a group of statements, depending on the value of an expression.

Create a new rule named "screw_part_number_rule". You can use the Keywords box on the Rule Syntax tab to assist you in filling in this rule.

```
Select Case port_a_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
                Screw_num1 = 100
        Case 1.25
                Screw_num1 = 125
        Case 1.50
                Screw_num1 = 150
        Case 2.00
                Screw_num1 = 200
        Case 2.50
                Screw_num1 = 250
        Case 3.00
                Screw_num1 = 300
End Select

port_a_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"
```

In this rule, we first compute a variable portion of the part number, based on the **port_a_size** parameter value. We hold this in a temporary placeholder variable named "Screw_num1". After that, we construct the entire part number string, which will look (for example) like this (for port_a_size = 0.50):

UNBRAKO-050-SCREW

We save this value in a parameter for later reference.

We need to do something similar for the Port B and C screws. To do this, copy and paste the rule text created above, and then replace “port_a_size” with “port_b_size” and “port_c_size”, respectively, and set the parameters “port_b_screw_part_number” and “port_c_screw_part_number”. This will result in the following additional rule text.

```
Select Case port_b_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
                Screw_num1 = 100
        Case 1.25
                Screw_num1 = 125
        Case 1.50
                Screw_num1 = 150
        Case 2.00
                Screw_num1 = 200
        Case 2.50
                Screw_num1 = 250
        Case 3.00
                Screw_num1 = 300
End Select

port_b_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"

Select Case port_c_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
                Screw_num1 = 100
        Case 1.25
                Screw_num1 = 125
        Case 1.50
                Screw_num1 = 150
        Case 2.00
                Screw_num1 = 200
        Case 2.50
                Screw_num1 = 250
        Case 3.00
                Screw_num1 = 300
End Select

port_c_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"
```

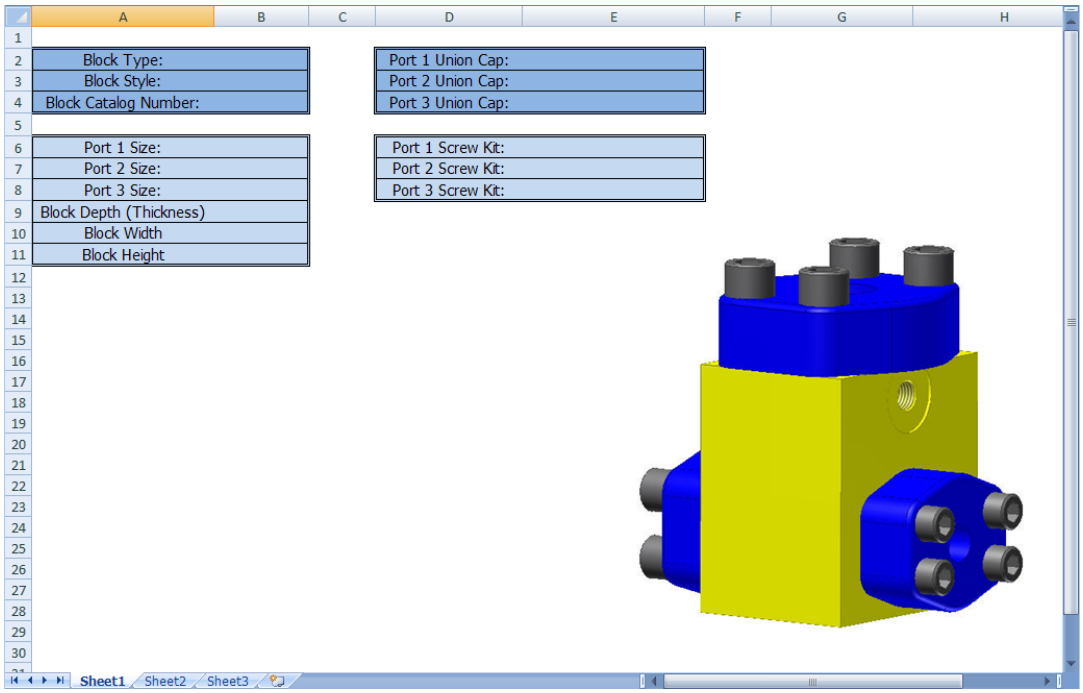
That's it -- click OK to end the rule.

WRITING INFORMATION TO AN EXCEL SPREADSHEET

Up to this point, we have been able to set all of the parameter values necessary to construct the model, and to compute the part numbers. The last rule we'll write in this

tutorial will pass the part numbers and other parameter values associated with this file to an Excel spreadsheet. Some of these part numbers are in the iParts, and the screw part number will be generated by a rule that we are going to write.

Below is an image of the spreadsheet that has been created for this tutorial:



CREATE UPDATE_EXCEL_SPREADSHEET_RULE

Add a new rule named “update_excel_spreadsheet_rule”. This rule will fill in the necessary cells dependent upon the state of the model.

For convenient reference, we have named all of the cells in the spreadsheet that we will be passing values to. These names loosely correspond to the information that will be written to each cell.

This rule utilizes a set of functions that are available from the Data Links category, found under the Rule Syntax tab in the iLogic Rule Editor. We’ll highlight these functions below.

The first portion of the rule opens the spreadsheet, and writes the first three cell values.

```
GoExcel.CellValue("part_number.xls", "Sheet1", "Block_Type") = component_type
GoExcel.CurrentCellValue("Block_Style") = block
GoExcel.CurrentCellValue("Block_Part_Number") = iProperties.Value("manifold_block:1",
```

```
"Project", "Part Number")
```

Here, we are referencing the "part_number.xls" spreadsheet file, which is included in this tutorial project. We then set the values for three named cells: **Block_Type**, **Block_Style**, and **Block_Part_Number**. The first two of these values are set from assembly parameters; the last is set from the block's Part Number iProperty.

The next portion of the rule writes the values of the three port sizes to the spreadsheet. Notice how we use a placeholder value of "N/A" if we're creating an elbow-style manifold.

```
GoExcel.CurrentCellValue("port_a_size") = port_a_size
If block = "tee" Then
        GoExcel.CurrentCellValue("port_b_size") = port_b_size
Else
        GoExcel.CurrentCellValue("port_b_size") = "N/A"
End If
GoExcel.CurrentCellValue("port_c_size") = port_c_size
```

The next few lines in the rule assign cell values from parameters contained in the manifold block component. Note the reference to the "manifold_block:1" component in this section.

```
GoExcel.CurrentCellValue("block_depth") = Parameter("manifold_block:1", "block_depth")
GoExcel.CurrentCellValue("block_width") = Parameter("manifold_block:1", "block_width")
GoExcel.CurrentCellValue("block_height") = Parameter("manifold_block:1", "block_height")
```

The next portion of the rule assigns cell values from the part numbers of the union component, as well as from the screw parts, as computed by **screw_part_number_rule**.

```
GoExcel.CurrentCellValue("port_a_union_cap") = port_a_union_part_number
GoExcel.CurrentCellValue("port_a_screw_kit") = port_a_screw_part_number
If block = "tee"Then
GoExcel.CurrentCellValue("port_b_union_cap") = port_b_union_part_number
GoExcel.CurrentCellValue("port_b_screw_kit") = port_b_screw_part_number
Else
GoExcel.CurrentCellValue("port_b_union_cap") = "N/A"
GoExcel.CurrentCellValue("port_b_screw_kit") = "N/A"
End If
GoExcel.CurrentCellValue("port_c_union_cap") = port_c_union_part_number
GoExcel.CurrentCellValue("port_c_screw_kit") = port_c_screw_part_number
```

As with **port_b_size** above, notice how there is conditional handling of the Port B-related values.

Finally, we need to save the changes to the spreadsheet. This is easily accomplished

with one function available in the Data Links category. Double-click “GoExcel.Save”.

GoExcel.Save

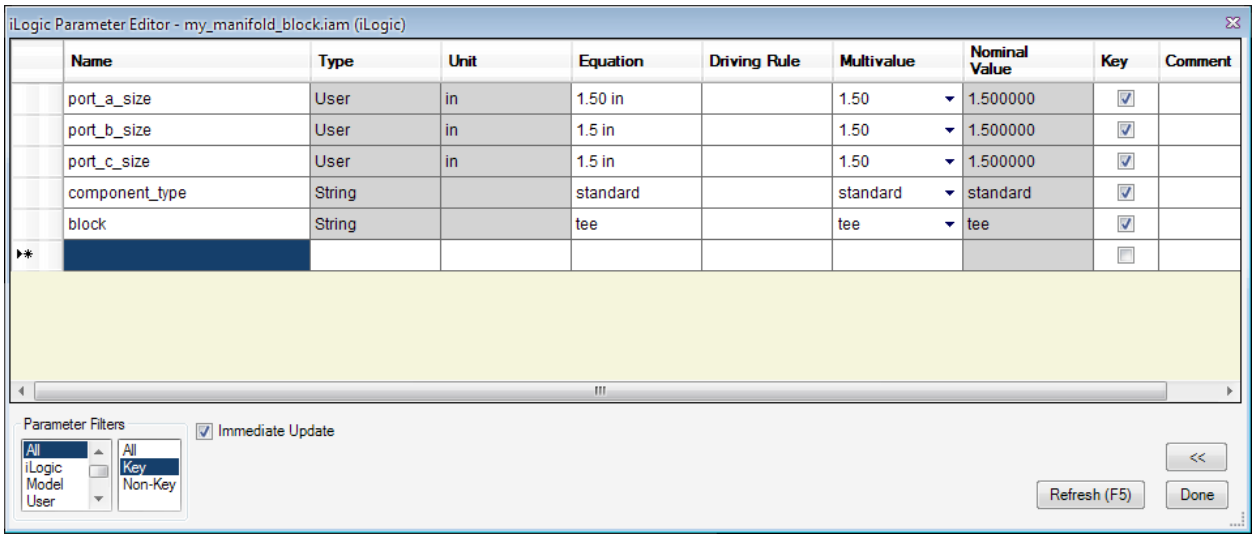
Now, press OK to save and close the rule. The spreadsheet will be updated as the rule fires upon closing. The rule will automatically close the spreadsheet when it is finished executing.

Now open the spreadsheet with Excel and see that it is up to date. Make sure that you close the spreadsheet document before the rule is executed again as the rule cannot update the spreadsheet if it is already opened with Excel.

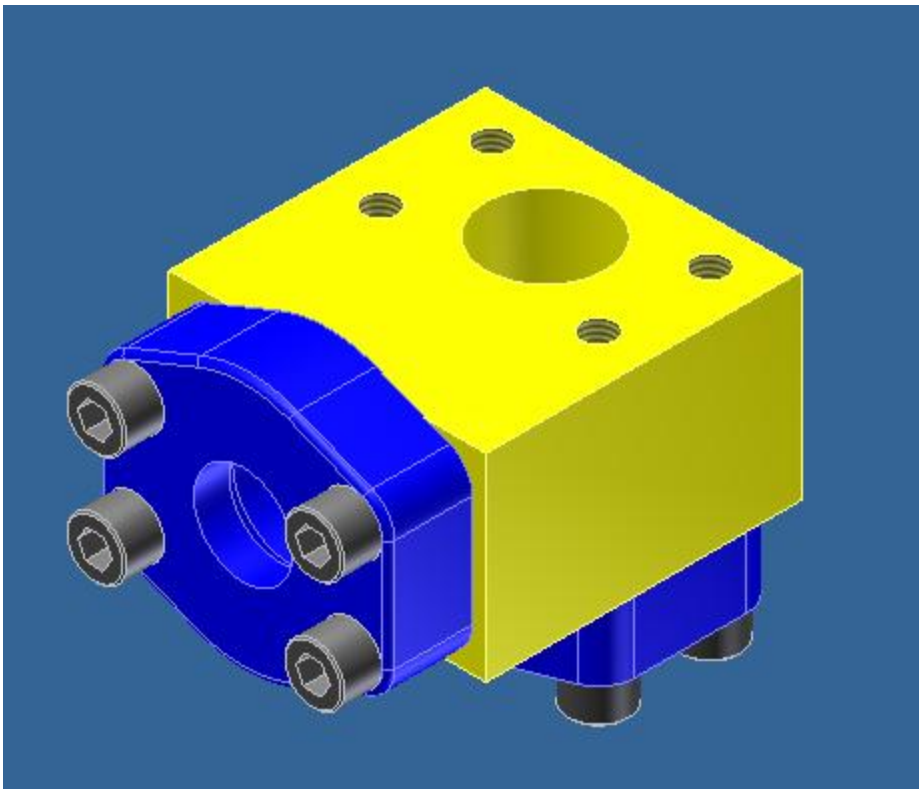
TEST YOUR RULES

Now that you’ve completed all the rules in your assembly, you can test them to see if all are working properly and producing the desired results. Testing can be done by changing the current parameter values using the iLogic Parameters dialog.

Click the iLogic Parameters icon in the iLogic panel bar or toolbar. From the Parameters dialog, select “Key” from the Parameter Filters list to view only the key parameters defined earlier in this tutorial:



Change the **block** parameter value from “tee” to “elbow” (click in some other cell or hit Tab to apply the change). Note how the model changes according to the rules:



Next, change **port_a_size** to some other value. Since **component_type** is set to "standard", the other two port sizes will also change, and the entire manifold size will change.

As these changes are made, the PartNumber.xls spreadsheet will also be updated. Open it with Excel for viewing and validation that the cell values reflect the current state of the model.

This completes this tutorial.

APPENDIX – FULL TEXT OF RULES

ASSEMBLY_TO_PARTS_RULE

```
Parameter("manifold_block:1", "block") = block
Parameter("manifold_block:1", "component_type") = component_type
Parameter("manifold_block:1", "port_a_size") = port_a_size
Parameter("manifold_block:1", "port_b_size") = port_b_size
Parameter("manifold_block:1", "port_c_size") = port_c_size
```

COMPONENT_TYPE_RULE

```
If component_type = "standard" Then
port_b_size = port_a_size
port_c_size = port_a_size
End If
```

PORT_A_RULE

```
i = iPart.FindRow("port_a_union", "port_size", "=", port_a_size)
port_a_y_dist_between_screws = iPart.CurrentRowValue("y_dist_betwn_screw")
port_a_x_dist_between_screws = iPart.CurrentRowValue("x_dist_betwn_screw")

If port_a_size = .50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-01")
ElseIf port_a_size = 0.75 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-02")
ElseIf port_a_size = 1.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-02")
ElseIf port_a_size = 1.25 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-03")
ElseIf port_a_size = 1.50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-04")
ElseIf port_a_size = 2.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-04")
ElseIf port_a_size = 2.50 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-05")
ElseIf port_a_size = 3.00 Then
        iPart.ChangeRow("port_a_union_screw", "Screw-06")
End If

port_a_union_part_number = iProperties.Value("port_a_union", "Project", "Part Number")
```

PORT_B_RULE

```
If block = "elbow" Then
	isTee = False
Else
	isTee = True
End If

Constraint.IsActive("port_b_cap_center") = isTee
Constraint.IsActive("port_b_cap_hole") = isTee
Constraint.IsActive("port_b_cap_face") = isTee
Constraint.IsActive("port_b_cap_screw") = isTee
Component.IsActive("port_b_union") = isTee
Component.IsActive("port_b_screw_pattern") = isTee

if isTee Then
	i = iPart.FindRow("port_b_union", "port_size", "=", port_b_size)
	port_b_y_dist_between_screws = iPart.CurrentRowValue("y_dist_betwn_screw")
	port_b_x_dist_between_screws = iPart.CurrentRowValue("x_dist_betwn_screw")
	port_b_union_part_number = iProperties.Value("port_b_union", "Project", "Part
Number")
End If

If port_b_size = .50 then
	iPart.ChangeRow("port_b_union_screw", "Screw-01")
elseif port_b_size = .75 then
	iPart.ChangeRow("port_b_union_screw", "Screw-02")
elseif port_b_size = 1.00 then
	iPart.ChangeRow("port_b_union_screw", "Screw-02")
elseif port_b_size = 1.25 then
	iPart.ChangeRow("port_b_union_screw", "Screw-03")
elseif port_b_size = 1.50 then
	iPart.ChangeRow("port_b_union_screw", "Screw-04")
elseif port_b_size = 2.00 then
	iPart.ChangeRow("port_b_union_screw", "Screw-04")
elseif port_b_size = 2.50 then
	iPart.ChangeRow("port_b_union_screw", "Screw-05")
elseif port_b_size = 3.00 then
	iPart.ChangeRow("port_b_union_screw", "Screw-06")
end if
```

PORT_C_RULE

```
i = iPart.FindRow("port_c_union", "port_size", "=", port_c_size)
port_c_y_dist_between_screws = iPart.CurrentRowValue("y_dist_betwn_screw")
port_c_x_dist_between_screws = iPart.CurrentRowValue("x_dist_betwn_screw")

If port_c_size = .50 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-01")
ElseIf port_c_size = 0.75 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-02")
ElseIf port_c_size = 1.00 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-02")
ElseIf port_c_size = 1.25 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-03")
ElseIf port_c_size = 1.50 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-04")
ElseIf port_c_size = 2.00 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-04")
ElseIf port_c_size = 2.50 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-05")
ElseIf port_c_size = 3.00 Then
        iPart.ChangeRow("port_c_union_screw", "Screw-06")
End If

port_c_union_part_number = iProperties.Value("port_c_union", "Project", "Part Number")
```

SCREW_PART_NUMBER_RULE

```
Select Case port_a_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
                Screw_num1 = 100
        Case 1.25
                Screw_num1 = 125
        Case 1.50
                Screw_num1 = 150
        Case 2.00
                Screw_num1 = 200
        Case 2.50
                Screw_num1 = 250
        Case 3.00
                Screw_num1 = 300
End Select

port_a_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"

Select Case port_b_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
                Screw_num1 = 100
        Case 1.25
                Screw_num1 = 125
        Case 1.50
                Screw_num1 = 150
        Case 2.00
                Screw_num1 = 200
        Case 2.50
                Screw_num1 = 250
        Case 3.00
                Screw_num1 = 300
End Select

port_b_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"

Select Case port_c_size
        Case .50
                Screw_num1 = 050
        Case .75
                Screw_num1 = 075
        Case 1.00
```

```
            Screw_num1 = 100
        Case 1.25
            Screw_num1 = 125
        Case 1.50
            Screw_num1 = 150
        Case 2.00
            Screw_num1 = 200
        Case 2.50
            Screw_num1 = 250
        Case 3.00
            Screw_num1 = 300
End Select

port_c_screw_part_number = "UNBRAKO-" & Screw_num1 & "-SCREW"
```

UPDATE_EXCEL_SPREADSHEET_RULE

```
GoExcel.CellValue("part_number.xls", "Sheet1", "Block_Type") = component_type
GoExcel.CurrentCellValue("Block_Style") = block
GoExcel.CurrentCellValue("Block_Part_Number") = iProperties.Value("manifold_block:1",
"Project", "Part Number")

GoExcel.CurrentCellValue("port_a_size") = port_a_size
If block = "tee" Then
        GoExcel.CurrentCellValue("port_b_size") = port_b_size
Else
        GoExcel.CurrentCellValue("port_b_size") = "N/A"
End If
GoExcel.CurrentCellValue("port_c_size") = port_c_size

GoExcel.CurrentCellValue("block_depth") = Parameter("manifold_block:1", "block_depth")
GoExcel.CurrentCellValue("block_width") = Parameter("manifold_block:1", "block_width")
GoExcel.CurrentCellValue("block_height") = Parameter("manifold_block:1", "block_height")

GoExcel.CurrentCellValue("port_a_union_cap") = port_a_union_part_number
GoExcel.CurrentCellValue("port_a_screw_kit") = port_a_screw_part_number
If block = "tee"Then
GoExcel.CurrentCellValue("port_b_union_cap") = port_b_union_part_number
GoExcel.CurrentCellValue("port_b_screw_kit") = port_b_screw_part_number
Else
GoExcel.CurrentCellValue("port_b_union_cap") = "N/A"
GoExcel.CurrentCellValue("port_b_screw_kit") = "N/A"
End If
GoExcel.CurrentCellValue("port_c_union_cap") = port_c_union_part_number
GoExcel.CurrentCellValue("port_c_screw_kit") = port_c_screw_part_number

GoExcel.Save
```