Sub CreateAndConfigureSump()

' This code creates a new copy of the sump assembly template and places it in our new folder structure

' It then updates the sump geometry based on values we pass to the assembly

Dim strNewSumpFilename As String

' This string represents the new name of our unique, copied sump assembly file

strNewSumpFilename = PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump Pipe Assy - " & PROJECT\_ID & ".iam"

' We first check to make sure the sump assembly file has not been previously created

' If it hasn't yet been created, we do a Windows Copy operation to make a new copy in our new folder

' We also make copies of the part files that will go into our sump assembly

If System.IO.File.Exists(strNewSumpFilename) = False Then

System.IO.File.Copy(TEMPLATE\_PATH & "Sump Assy\Sump Pipe Assy.iam", PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump Pipe Assy - " & PROJECT\_ID & ".iam")

System.IO.File.Copy(TEMPLATE\_PATH & "Sump Assy\Sump-Angled Pipe.ipt", PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump-Angled Pipe - " & PROJECT\_ID & ".ipt")

System.IO.File.Copy(TEMPLATE\_PATH & "Sump Assy\Sump-Straight Pipe.ipt", PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump-Straight Pipe - " & PROJECT\_ID & ".ipt")

' This here is a sample of how to make a variable that represents an assembly document

' It uses the Inventor API, which you can use freely (for the most part) throughout iLogic rules

Dim oSumpAssy As Inventor.AssemblyDocument

' This statement tells oSumpAssy to represent the newly created file, and opens it up in the Inventor interface

oSumpAssy = ThisApplication.Documents.Open(strNewSumpFilename, True)

' When the copied sump assembly initially opens, it will reference the old part files in our template folder

' We need to change that so that the newly copied angle and straight pipe files are referenced by the assembly

' The following code uses the Inventor API functionality to do that

' This is similar to selecting the "Replace Components" command in the Inventor application

Dim strOldAnglePipe, strNewAnglePipe As String

strOldAnglePipe = TEMPLATE\_PATH & "Sump Assy\Sump-Angled Pipe.ipt"

strNewAnglePipe = PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump-Angled Pipe - " & PROJECT\_ID & ".ipt"

oSumpAssy.File.ReferencedFileDescriptors.Item(strOldAnglePipe).ReplaceReference(strNewAnglePipe)

Dim strOldStraightPipe, strNewStraightPipe As String

strOldStraightPipe = TEMPLATE\_PATH & "Sump Assy\Sump-Straight Pipe.ipt"

strNewStraightPipe = PROJECT\_PATH & PROJECT\_ID & "\Sump Assy\Sump-Straight Pipe - " & PROJECT\_ID & ".ipt"

oSumpAssy.File.ReferencedFileDescriptors.Item(strOldStraightPipe).ReplaceReference(strNewStraightPipe)

' Once we've updated the file references in the sump assembly file, we can save and then close it

oSumpAssy.Save

oSumpAssy.Close

End If

' The sump assembly was created in the exact same orientation as our master tank assembly

' This means we don't need to rotate the sump assembly when placing it into the master tank assembly

' That means we don't need a matrix, but can just define a point (X, Y, Z coordinates) of where to place it

Dim pointO = ThisDoc.Geometry.Point(0, -TANK\_OD / 2 + SUMP\_H, TANK\_L / 2)

' This can be taken from an iLogic snippet, and is used to insert components into assemblies

' This code inserts our newly created sump pipe assembly into our master tank assembly file

' Instead of placing at the origin, it places it based on our input point we created (pointO)

' Note that we are grounding all geometry, and we are not using any constraints to place the assembly

Dim componentO = Components.Add("Sump Pipe Assembly:1", strNewSumpFilename, position := pointO, grounded := True, visible := True, appearance := Nothing)

' These statements pass parameters from our master assembly file into the sump pipe assembly file

Parameter("Sump Pipe Assembly:1", "PROJECT\_ID") = PROJECT\_ID

Parameter("Sump Pipe Assembly:1", "PROJECT\_PATH") = PROJECT\_PATH

Parameter("Sump Pipe Assembly:1", "SUMP\_SIZE") = SUMP\_SIZE

Parameter("Sump Pipe Assembly:1", "SUMP\_H") = SUMP\_H

Parameter("Sump Pipe Assembly:1", "SUMP\_PIPE\_PROJ") = SUMP\_PIPE\_PROJ

Parameter("Sump Pipe Assembly:1", "TANK\_OD") = TANK\_OD

Parameter("Sump Pipe Assembly:1", "DISH\_DEPTH") = DISH\_DEPTH

' Once all the parameters are updated in the sump pipe assembly file, we want to run its update rule

' This will allow the sump pipe assembly to update all its own parts and components itself

iLogicVb.RunRule("Sump Pipe Assembly:1", "Update Children Parts")

' Once the sump pipe assembly is created and placed, it still needs an end connection

' This uses the "GetFlangeFilename" function (near the bottom of this rule)

' It will automatically determine the filename based on flange type, flange end connection, and drain size

Dim strFlangeName As String = GetFlangeFilename(SUMP\_FL\_TYPE, SUMP\_FL\_END, SUMP\_SIZE)

' This uses the "GetFrontOrRearMatrix" function (near the bottom of this rule)

' This will automatically determine the location matrix based on several factors

Dim matrixP = GetFrontOrRearMatrix(SUMP\_FL\_TYPE, SUMP\_FL\_END, SUMP\_H, SUMP\_PIPE\_PROJ, SUMP\_SIZE, "Front", "Bottom")

' This can be taken from an iLogic snippet, and is used to insert components into assemblies

' This code inserts the selected end connection part or assembly into our master tank assembly file

' Instead of placing at the origin, it places it based on our input matrix we created (matrixP)

' Note that we are grounding all geometry, and we are not using any constraints to place the end connection

Dim componentP = Components.Add("Sump Valve:1", strFlangeName, position := matrixP, grounded := True, visible := True, appearance := Nothing)

End Sub