Sub CreateAndConfigureSkid()

' This code creates a new copy of the skid assembly template in our new folder structure

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

Dim strNewSkidFilename As String

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

strNewSkidFilename = PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Skid Assy - " & PROJECT\_ID & ".iam"

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

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

' 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 subAssy1 As AssemblyDocument

' This code uses the "CopyComponents" subroutine (see above) to copy the skid assembly, and all its children

' This will not work if you don't want some of the parts in the assembly to have unique copies

' The function also changes the references in the skid assembly to point to the newly created part files

CopyComponents(TEMPLATE\_PATH & "Skid Assy\", "Skid Assy.iam", "Skid Assy")

' This statement tells subAssy1 to represent the template file, and opens it up in the Inventor interface

subAssy1 = ThisApplication.Documents.Open(strNewSkidFilename, True)

' This statement changes the occurrence names of the existing skid components in the model browser

' This will allow the rule that passes parameters in our skid sub-assembly to still work

subAssy1.ComponentDefinition.Occurrences(1).Name = "Skid-1:1"

subAssy1.ComponentDefinition.Occurrences(2).Name = "Skid-1:2"

subAssy1.Save

subAssy1.Close

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

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

Dim componentB = Components.Add("Skid Assy:1", strNewSkidFilename, position := Nothing, grounded := True, visible := True, appearance := Nothing)

' Change our flange radius if the TANK\_OD is 30" or less

If TANK\_OD <= 30 in Then SKID\_FLG\_RAD = .1 in

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

Parameter("Skid Assy:1", "PROJECT\_ID") = PROJECT\_ID

Parameter("Skid Assy:1", "PROJECT\_PATH") = PROJECT\_PATH

Parameter("Skid Assy:1", "TANK\_OD") = TANK\_OD

Parameter("Skid Assy:1", "TANK\_L") = TANK\_L

Parameter("Skid Assy:1", "SHELL\_W\_1") = SHELL\_W\_1

Parameter("Skid Assy:1", "SHELL\_W\_2") = SHELL\_W\_2

Parameter("Skid Assy:1", "SHELL\_Q\_1") = SHELL\_Q\_1

Parameter("Skid Assy:1", "SHELL\_Q\_2") = SHELL\_Q\_2

Parameter("Skid Assy:1", "SKID\_FW") = SKID\_FW

Parameter("Skid Assy:1", "SKID\_FH") = SKID\_FH

Parameter("Skid Assy:1", "SKID\_FL\_THK") = SKID\_FL\_THK

Parameter("Skid Assy:1", "SKID\_FLG\_RAD") = SKID\_FLG\_RAD

Parameter("Skid Assy:1", "SKID\_WEB\_THK") = SKID\_WEB\_THK

Parameter("Skid Assy:1", "SKID\_BEND\_L") = SKID\_BEND\_L

Parameter("Skid Assy:1", "SKID\_ROD\_D") = SKID\_ROD\_D

Parameter("Skid Assy:1", "DISH\_DEPTH") = DISH\_DEPTH

' Once all the parameters are updated in the skid assembly file, we want to run its creation rule

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

iLogicVb.RunRule("Skid Assy:1", "Create Skid")

End If

End Sub