Sub Main()

' This function contains the required code to create the skid assembly based on parameters that are passed from the parent assembly

' If the diameter is 30" or less, change out I-beam skids for rectangular beams

If TANK\_OD <= 30 Then

Dim oSkidAssy As AssemblyDocument

' This sets a reference to this file - the skid assembly

oSkidAssy = ThisApplication.Documents.ItemByName(PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Skid Assy - " & PROJECT\_ID & ".iam")

' We now have string variables the represent the current skids, the new skids, and the template file we need to copy for the

' new skids

Dim strOldFileName, strNewFileName, strTemplateFileName As String

strOldFileName = PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Skid-1 - " & PROJECT\_ID & ".ipt"

strNewFileName = PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Skid-2 - " & PROJECT\_ID & ".ipt"

strTemplateFileName = TEMPLATE\_PATH & "\Skid Assy\Skid-2.ipt"

' These commands copy the new skid (rectangular tube) beam to our project folder, and then update the reference

' in this assembly file ("Component Replace") to the new skid tube model

' It then saves this assembly file, and gets rid of our reference to it

System.IO.File.Copy(strTemplateFileName, strNewFileName)

oSkidAssy.File.ReferencedFileDescriptors.Item(strOldFileName).ReplaceReference(strNewFileName)

oSkidAssy.Save

oSkidAssy = Nothing

End If

' Create a fresh copy of the saddle assembly, and add it to the assembly at the origin

Dim strNewSaddleFilename As String

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

' We check to make sure a saddle assembly hasn't been previously created

' If it hasn't been created previously, then we create a new copy of it

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

Dim subAssy1 As AssemblyDocument

Dim intTotalShells As Integer

Dim intConstraintNumber As Integer

' This uses our "CopyComponents" subroutine (found at the end of this rule) to do a kind of "Copy Design" of the

' saddle assembly

' This gives us a new copy of it that can be modified without affecting the template files

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

' This will tell us how many shell plates are needed for this sepcific tank

' We use this to determine how many saddle assemblies will need to be placed into the assembly

' We need a saddle under each seam for proper support

intTotalShells = SHELL\_Q\_1 + SHELL\_Q\_2

subAssy1 = ThisApplication.Documents.ItemByName(strNewSaddleFilename)

' If OD of the tank is 30" or less, sub the saddle part-2 with the alternate part file (SK-Saddle-2b.ipt)

' This is a separate part that is needed if we have to use rectangular tubing instead of I-beams

If TANK\_OD <= 30 in Then

' This code will copy the new saddle support piece into our project folder, and substitute the assembly

' reference ("Component Replace") to the new plate

Dim strOldSaddle, strNewSaddle As String

strOldSaddle = PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Saddle-2a - " & PROJECT\_ID & ".ipt"

strNewSaddle = PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Saddle-2b - " & PROJECT\_ID & ".ipt"

System.IO.File.Copy(TEMPLATE\_PATH & "\Skid Assy\Saddle-2b.ipt", strNewSaddle)

subAssy1.File.ReferencedFileDescriptors.Item(strOldSaddle).ReplaceReference(strNewSaddle)

subAssy1.Save

End If

' Change Occurrence names of the saddle components in the model browser so that the

' "Update Saddle Parameters" rule will work As-Is

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

subAssy1.ComponentDefinition.Occurrences(3).Name = "Saddle-2:1"

subAssy1.ComponentDefinition.Occurrences(4).Name = "Saddle-3:1"

' Each constraint that is created needs a unique identifier

' Since we already have 4 constraints that have been created in the model template, we want to start numbering the new constraints we

' create with the number 5

intConstraintNumber = 5

' We need one more saddle assembly than number of shells (i.e. if we have 5 shell plates, there are 6 seams that need to be supported)

' This For loop will first add a new instance of the saddle assembly into our skid assembly model

' Then flush constraints will be created for the new saddle assembly for the "YZ" and "XZ" planes

' After that, the total offset for the current seam will be calculated, and the saddle assemblies last flush constraint will have

' an offset so that they are spaced perfectly under each seam

For i = 1 To intTotalShells + 1

Dim componentA = Components.Add("Saddle Assy:" & i, PROJECT\_PATH & PROJECT\_ID & "\Skid Assy\Saddle Assy - " & PROJECT\_ID & ".iam")

Constraints.AddFlush("Flush:" & intConstraintNumber, "", "YZ Plane",

"Saddle Assy:" & i, "YZ Plane")

Constraints.AddFlush("Flush:" & intConstraintNumber + 1, "", "XZ Plane",

"Saddle Assy:" & i, "XZ Plane")

Dim intSaddleOffset As Integer

If i > SHELL\_Q\_1 Then

intSaddleOffset = SHELL\_Q\_1 \* SHELL\_W\_1 + (i - SHELL\_Q\_1 - 1) \* SHELL\_W\_2

Else

intSaddleOffset = (i - 1) \* SHELL\_W\_1

End If

Constraints.AddFlush("Flush:" & intConstraintNumber + 2, "Saddle Assy:" & i,

"XY Plane", "", "Work Plane1", intSaddleOffset)

intConstraintNumber = intConstraintNumber + 3

Next

' After we have placed all of the instances of the saddle assembly in our skid assembly, and located them properly, we then need

' to update the sizes of the saddles, and we do that by running the "Update Saddle Parameters" rule

iLogicVb.RunRule("Update Saddle Parameters")

subAssy1.Close

End If

End Sub

Sub CopyComponents(strFilePath As String, strAssemblyName As String, strFolderName As String)

' This is the same subroutine used in the top level "Create New Tank.iam" file

' For more information regarding this subroutine, refer to the notes in the top assembly level file

Dim oAsmDoc As AssemblyDocument

oAsmDoc = ThisApplication.Documents.Open(strFilePath & strAssemblyName, True)

oAsmDoc.SaveAs(PROJECT\_PATH & PROJECT\_ID & "\" & strFolderName & "\" & Left(strAssemblyName, strAssemblyName.Length - 4) & " - " & PROJECT\_ID & ".iam", False)

Dim oRefDocs As DocumentsEnumerator

oRefDocs = oAsmDoc.AllReferencedDocuments

Dim oRefDoc As Document

For Each oRefDoc In oRefDocs

Dim strNewFileName As String

Dim strOldFileName As String

strOldFileName = strFilePath & oRefDoc.DisplayName

strNewFileName = PROJECT\_PATH & PROJECT\_ID & "\" & strFolderName & "\" & Left(oRefDoc.DisplayName, oRefDoc.DisplayName.Length - 4) & " - " & PROJECT\_ID & ".ipt"

oRefDoc.SaveAs(strNewFileName, True)

oAsmDoc.File.ReferencedFileDescriptors.Item(strOldFileName).ReplaceReference(strNewFileName)

Next

End Sub