

Note: This iLogic code is linked from Luke Davenport's blog on Cadline Community here (June 2013).

Note when copying and pasting this code, all gaps between lines must be removed otherwise errors will occur when code is run.

<http://www.cadlinecommunity.co.uk/Blogs/LukeDavenport/Default.aspx>

'Create Contains' iLogic Rule

'Start of iLogic code

'Note to self – subscribe to Luke Davenport's blog and follow him on

'twitter @LukeCadline

'Return view to Home view

ThisApplication.CommandManager.ControlDefinitions.Item _

("AppViewCubeHomeCmd").Execute

'define current document

Dim openDoc As Document

openDoc = ThisDoc.Document

Dim oAsmCompDef As AssemblyComponentDefinition

oAsmCompDef = openDoc.ComponentDefinition

Dim oViewRep As DesignViewRepresentation

Try

'Activate a writeable View Rep (master view rep is not writeable)

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").activate

Catch

'Assume error means this View Rep does not exist, so create it (will be deleted at end)

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("WIP View Rep")

End Try

'Delete all existing 'Contains' View reps

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations

If oViewRep.Name.Contains("Contains") Then

oViewRep.Delete

End If

Next

Start:

iCount = 1

'request part names to create view reps for.

Do Until iCount = 1000

StrInput1 = InputBox("Enter Part Names to Add to View Rep" & vbCrLf & vbCrLf & _

"Enter c to cancel", "Set View Representations", "Enter all or part of PN", MessageBoxButtons.Button1)

If StrInput1 = "c" Then

Return

End If

'Check to see whether that view rep has already been created

ViewRepExists = 0

```

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations
If oViewRep.Name.Contains("Contains " & StrInput1 & "") Then
ViewRepExists = 1

oCreatedAlready = MessageBox.Show("That View Rep Has Already Been Created!" & vbCrLf & vbCrLf &
"Do You Still Want To Continue?","Already Entered",MessageBoxButtons.YesNo)

If oCreatedAlready = vbYes Then
Goto Start

Else

Return

End If

End If

Next

If ViewRepExists = 0 Then
'Create new View Rep

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("Contains " &
StrInput1)

'Activate new View Rep

oViewRep.activate

End If

oOccCounter = 0

oSubOccCounter = 0

oSubSubOccCounter = 0

'look at all of the components in the assembly

Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition

'define the first level components collection

Dim oCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubCompOcc As Inventor.ComponentOccurrence

```

'define the next level components collection

Dim oSubSubCompOcc As Inventor.ComponentOccurrence

'Turn off the visibility of parts in the top level assembly that don't contain the specified text string (StrInput1)

For Each oCompOcc in oCompDef.Occurrences

 If oCompOcc.Suppessed = False Then

 If oCompOcc.Name.Contains(StrInput1) Then

 oCompOcc.Visible = True

 'Increment counter for top level components

 oOccCounter = oOccCounter+1

 Else

 oCompOcc.Visible = False

 End If

'Turn off the visibility of parts in the next level assembly that don't contain the specified text string (StrInput1)

 For Each oSubCompOcc In oCompOcc.SubOccurrences

 If oSubCompOcc.Suppessed = False Then

 If oSubCompOcc.Name.Contains(StrInput1) Then

 oSubCompOcc.Visible = True

 'Increment counter for first level components

 oSubOccCounter = oSubOccCounter+1

 Else

 oSubCompOcc.Visible = False

 End If

 End If

 For Each oSubSubCompOcc In oSubCompOcc.SubOccurrences

 If oSubSubCompOcc.Suppessed = False Then

 If oSubSubCompOcc.Name.Contains(StrInput1) Then

 oSubSubCompOcc.Visible = True

 'Increment counter for first level components

 oSubSubOccCounter = oSubSubOccCounter+1

```

        Else
            oSubSubCompOcc.Visible = False
        End If
    End If

    Next

Next

End If

Next

Next

'Rename View Rep to include component counts
oViewRep.Name = ("Contains " & StrInput1 & " (Qty: " & oOccCounter & " Top Level, " &
oSubOccCounter & " Sub Comps, " _
& oSubSubOccCounter & " Sub Sub Comps)")

'lock the new view rep
oViewRep.Locked = True

'Count no. of view reps already created
ViewRepCount = -1

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations
If oViewRep.Name.Contains("Contains") Then
ViewRepCount = ViewRepCount + 1
End If
Next

'Zoom all
ThisApplication.ActiveView.Fit

'See if another View Rep is required.
oContinue = MessageBox.Show("Add Another View Rep?" & vbLf & ViewRepCount & " View Reps
Have Been Created", "Continue Creating?", MessageBoxButtons.YesNo)

If oContinue = vbYes Then

```

```
iCount = iCount+1
```

```
Else
```

```
iCount = 1000
```

```
End If
```

```
Loop
```

```
Try
```

```
'Delete WIP View Rep
```

```
oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").delete
```

```
Catch
```

```
End Try
```

Check For Update 'Contains' iLogic Rule

```
'Check for Update - Contains' Rule;
```

```
'Note I must subscribe to Luke Davenport's blog – it'll change my life.
```

```
'http://www.cadlinecommunity.co.uk/Blogs/lukedavenport/Default.aspx
```

```
'And if I am a twitter user I really must follow him.
```

```
'define current document
```

```
Dim openDoc As Document
```

```
openDoc = ThisDoc.Document
```

```
Dim oAsmCompDef As AssemblyComponentDefinition
```

```
oAsmCompDef = openDoc.ComponentDefinition
```

```
Dim oViewRep As DesignViewRepresentation
```

```
'loop through each view representation in the assembly
```

```
For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations
```

```
    If oViewRep.Name.Contains("Contains") Then
```

```
        'Set initial values for component counts
```

```
        ActualOccCounter = 0
```

```
        ActualSubOccCounter = 0
```

```
            ActualSubSubOccCounter = 0
```

```
        'Find current value for Top level component visibility count
```

```
        CurrentOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.IndexOf(":")-2))
```

```
        'Find current value for Sub component visibility count
```

```
        CurrentSubOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.IndexOf(",")-2))
```

```
        'Find current value for Sub Sub component visibility count
```

```
        CurrentSubSubOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.IndexOf("s,")-3))
```

```
        'Create string for current contents of view rep (this is located inside the view rep name so needs to  
be pulled out)
```

```
        CurrentRepName = Mid(oViewRep.Name,11,Len(oViewRep.Name)-12-((Len(oViewRep.Name))-  
((oViewRep.Name.IndexOf("("))))))
```

```
        'look at all of the components in the assembly
```

```

Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition
'define the first level components collection

Dim oCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubCompOcc As Inventor.ComponentOccurrence
    'define the next level components collection

Dim oSubSubCompOcc As Inventor.ComponentOccurrence

'Check no. of parts in the top level assembly that contain the current view rep name
For Each oCompOcc in oCompDef.Occurrences
    If oCompOcc.Suppessed = False Then
        If oCompOcc.Name.Contains(CurrentRepName) Then
            ActualOccCounter = ActualOccCounter+1
        End If
    End If

'Check no. of parts in the next level assembly that contain the current view rep name
For Each oSubCompOcc In oCompOcc.SubOccurrences
    If oSubCompOcc.Suppessed = False Then
        If oSubCompOcc.Name.Contains(CurrentRepName) Then
            ActualSubOccCounter = ActualSubOccCounter+1
        End If
    End If

    For Each oSubSubCompOcc In
oSubCompOcc.SubOccurrences
        If oSubSubCompOcc.Suppessed = False Then
            If oSubSubCompOcc.Name.Contains(CurrentRepName) Then
                ActualSubSubOccCounter = ActualSubSubOccCounter+1
            End If
        End If
    Next

```

Next

End If

Next

'Compare actual component counts with the current view rep counts

TestEqual =

String.Compare(CStr(ActualOccCounter),CStr(Left(CurrentOccCounter,Len(CStr(ActualOccCounter))))
,True)

'Compare actual subcomponent counts with the current view rep counts

SubTestEqual =

String.Compare(CStr(ActualSubOccCounter),CStr(Left(CurrentSubOccCounter,Len(CStr(ActualSubOcc
Counter))))),True)

'Compare actual subsubcomponent counts with the current view rep counts

SubSubTestEqual =

String.Compare(CStr(ActualSubSubOccCounter),CStr(Left(CurrentSubSubOccCounter,Len(CStr(Actual
SubSubOccCounter))))),True)

'if counts don't match then inform user and allow update rules to be run

If TestEqual <> 0 Or SubTestEqual <> 0 Or SubSubTestEqual <> 0 Then

UpdateNow = MessageBox.Show("The View Rep Containing "" & CurrentRepName & "" Needs
Updating" _

& vbCrLf & vbCrLf & "Update All 'Contains' View Reps Now?","iLogic View
Representations",MessageBoxButtons.YesNo)

If UpdateNow = vbYes Then

iLogicVb.RunRule("Update Contains")

End If

Return

End If

End If

Next

Update 'Contains' iLogic Rule

'Start of iLogic code

'Note to self – subscribe to Luke Davenport's blog and follow him on

'twitter @LukeCadline

'define current document

Dim openDoc As Document

openDoc = ThisDoc.Document

Dim oAsmCompDef As AssemblyComponentDefinition

oAsmCompDef = openDoc.ComponentDefinition

Dim oViewRep As DesignViewRepresentation

Try

'Activate a writeable View Rep (master view rep is not writeable)

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").activate

Catch

'Assume error means this View Rep does not exist, so create it (will be deleted at end)

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("WIP View Rep")

End Try

i = 0

'Populate ExistingViewRepsArray from existing view reps

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations

 If oViewRep.Name.Contains("Contains") Then

 oViewRep.Activate

 oViewRep.Locked = False

 i = i+1

'Create string for current contents of view rep (this is located inside the view rep name so needs to be pulled out.

 CurrentRepName = Mid(oViewRep.Name,11,Len(oViewRep.Name)-12-((Len(oViewRep.Name))-((oViewRep.Name.IndexOf("("))))))

oOccCounter = 0

oSubOccCounter = 0

oSubSubOccCounter = 0

'look at all of the components in the assembly

Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition

'define the first level components collection

Dim oCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubSubCompOcc As Inventor.ComponentOccurrence

'Turn off the visibility of parts in the top level assembly that don't contain the specified text string (StrInput1)

For Each oCompOcc in oCompDef.Occurrences

 If oCompOcc.Suppresse = False Then

 If oCompOcc.Name.Contains(CurrentRepName) Then

 oCompOcc.Visible = True

 oOccCounter = oOccCounter+1

 Else

 oCompOcc.Visible = False

 End If

'Turn off the visibility of parts in the next level assembly that don't

'contain the specified text String (StrInput1)

For Each oSubCompOcc In oCompOcc.SubOccurrences

 If oSubCompOcc.Suppresse = False Then

 If oSubCompOcc.Name.Contains(CurrentRepName) Then

 oSubCompOcc.Visible = True

 oSubOccCounter = oSubOccCounter+1

 Else

 oSubCompOcc.Visible = False

 End If

 End If

 For Each oSubSubCompOcc In oSubCompOcc.SubOccurrences

 If oSubSubCompOcc.Suppresse = False Then

 If oSubSubCompOcc.Name.Contains(CurrentRepName) Then

 oSubSubCompOcc.Visible = True

 oSubSubOccCounter = oSubSubOccCounter+1

 Else

 oSubSubCompOcc.Visible = False

 End If

 End If

Next

Next

End If

Next

'Rename View Rep to include component counts

oViewRep.Name = ("Contains '" & CurrentRepName & "' (Qty: " & oOccCounter & " Top Level, " & oSubOccCounter & " Sub Comps, " _

& oSubSubOccCounter & " Sub Sub Comps)")

'lock the new view rep

oViewRep.Locked = True

End If

Next

Try

'Delete WIP view rep

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").delete

Catch

End Try

'Return view to Home view

ThisApplication.CommandManager.ControlDefinitions.Item("AppViewCubeHomeCmd").Execute

If i=1 Then

MessageBox.Show(i & " View Rep Has Been Updated", "Success!")

Else If i>1 Then

MessageBox.Show(i & " View Reps Have Been Updated", "Success!")

Else

MessageBox.Show("No 'Contains' View Reps Exist" & vbLf & vbLf & "Please Create View Reps First!", "Failed")

End If

Create 'Begins' iLogic Rule

'Start of iLogic code

'Note to self – subscribe to Luke Davenport's blog and follow him on

'twitter @LukeCadline

'Return view to Home view

ThisApplication.CommandManager.ControlDefinitions.Item _

("AppViewCubeHomeCmd").Execute

'define current document

Dim openDoc As Document

openDoc = ThisDoc.Document

Dim oAsmCompDef As AssemblyComponentDefinition

oAsmCompDef = openDoc.ComponentDefinition

Dim oViewRep As DesignViewRepresentation

Try

'Activate a writeable View Rep (master view rep is not writeable)

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").activate

Catch

'Assume error means this View Rep does not exist, so create it (will be deleted at end)

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("WIP View Rep")

End Try

```

'Delete all existing 'Begins' View reps
For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations
If oViewRep.Name.Contains("Begins") Then
oViewRep.Delete
End If
Next

Start:

iCount = 1

'request part names to create view reps for.
Do Until iCount = 1000
StrInput1 = InputBox("Enter Beginning of Part Names to Add to View Rep" & vbCrLf & vbCrLf & _
"Enter c to cancel", "Set View Representations", "Enter beginning of
PN",MessageBoxDefaultButton.Button1)
If StrInput1 = "c" Then
Return
End If

'Check to see whether that view rep has already been created
For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations
If oViewRep.Name.Contains("Begins '" & StrInput1 & "'") Then
ViewRepExists = 1
oCreatedAlready = MessageBox.Show("That View Rep Has Already Been Created!" & vbCrLf & vbCrLf &
"Do You Still Want To Continue?", "Already Entered",MessageBoxButtons.YesNo)
If oCreatedAlready = vbYes Then
Goto Start
Else
Return

```

End If

End If

Next

If ViewRepExists = 0 Then

'Create new View Rep

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("Begins " &
StrInput1)

'Activate new View Rep

oViewRep.activate

End If

oOccCounter = 0

oSubOccCounter = 0

oSubSubOccCounter = 0

'look at all of the components in the assembly

Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition

'define the first level components collection

Dim oCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubSubCompOcc As Inventor.ComponentOccurrence

'Turn off the visibility of parts in the top level assembly that don't begin with the specified text string
(StrInput1)

For Each oCompOcc in oCompDef.Occurrences

 If oCompOcc.Suppessed = False Then

 If Left(oCompOcc.Name,Len(StrInput1)) = StrInput1 Then

 oCompOcc.Visible = True

 'Increment counter for top level components

```

oOccCounter = oOccCounter+1

Else

oCompOcc.Visible = False

End If

'Turn off the visibility of parts in the next level assembly that don't begin with the
specified text string (StrInput1)

For Each oSubCompOcc In oCompOcc.SubOccurrences

If oSubCompOcc.Suppresse = False Then

If Left(oSubCompOcc.Name,Len(StrInput1)) = StrInput1

Then

oSubCompOcc.Visible = True

'Increment counter for first level components

oSubOccCounter = oSubOccCounter+1

Else

oSubCompOcc.Visible = False

End If

End If

For Each oSubSubCompOcc In oSubCompOcc.SubOccurrences

If oSubSubCompOcc.Suppresse = False Then

If Left(oSubSubCompOcc.Name,Len(StrInput1)) =

StrInput1 Then

oSubSubCompOcc.Visible = True

'Increment counter for second level components

oSubSubOccCounter = oSubSubOccCounter+1

Else

oSubSubCompOcc.Visible = False

End If

End If

Next

Next

End If

```

Next

'Rename View Rep to include component counts

oViewRep.Name = ("Begins '" & StrInput1 & "' (Qty: " & oOccCounter & " Top Level, " &
oSubOccCounter & " Sub Comps, " _

& oSubSubOccCounter & " Sub Sub Comps)")

'lock the new view rep

oViewRep.Locked = True

'Count no. of view reps already created

ViewRepCount = -1

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations

If oViewRep.Name.Contains("Begins") Then

ViewRepCount = ViewRepCount + 1

End If

Next

'Zoom all

ThisApplication.ActiveView.Fit

'See if another View Rep is required.

oContinue = MessageBox.Show("Add Another View Rep?" & vbCrLf & ViewRepCount & " View Reps
Have Been Created", "Continue Creating?", MessageBoxButtons.YesNo)

If oContinue = vbYes Then

iCount = iCount+1

Else

iCount = 1000

End If

Loop

Try

'Delete WIP View Rep

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").delete

Catch

End Try

Check 'Begins' iLogic Rule

'Start of iLogic code

'Note to self – subscribe to Luke Davenport's blog and follow him on

'twitter @LukeCadline

'define current document

Dim openDoc As Document

openDoc = ThisDoc.Document

Dim oAsmCompDef As AssemblyComponentDefinition

oAsmCompDef = openDoc.ComponentDefinition

Dim oViewRep As DesignViewRepresentation

'loop through each view representation in the assembly

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations

 If oViewRep.Name.Contains("Begins") Then

 'Set initial values for component counts

 ActualOccCounter = 0

 ActualSubOccCounter = 0

 ActualSubSubOccCounter = 0

'Find current value for Top level component visibility count

```
CurrentOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.Indexof(":")-2))
```

'Find current value for Sub component visibility count

```
CurrentSubOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.Indexof(",")-2))
```

'Find current value for Sub Sub component visibility count

```
CurrentSubSubOccCounter = Right(oViewRep.Name,((Len(oViewRep.Name))-  
oViewRep.Name.Indexof("s")-2))
```

'Create string for current contents of view rep (this is located inside the view rep name so needs to be pulled out)

```
CurrentRepName = Mid(oViewRep.Name,9,Len(oViewRep.Name)-10-((Len(oViewRep.Name))-  
((oViewRep.Name.Indexof("("))))))
```

'look at all of the components in the assembly

```
Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition
```

'define the first level components collection

```
Dim oCompOcc As Inventor.ComponentOccurrence
```

'define the next level components collection

```
Dim oSubCompOcc As Inventor.ComponentOccurrence
```

'define the next level components collection

```
Dim oSubSubCompOcc As Inventor.ComponentOccurrence
```

'Check no. of parts in the top level assembly that contain the current view rep name

```
For Each oCompOcc in oCompDef.Occurrences
```

```
    If oCompOcc.Suppressed = False Then
```

```
If Left(oCompOcc.Name,Len(CurrentRepName)) = CurrentRepName Then
```

```
ActualOccCounter = ActualOccCounter+1
```

```
End If
```

```
'Check no. of parts in the next level assembly that contain the current view rep name
```

```
For Each oSubCompOcc In oCompOcc.SubOccurrences
```

```
    If oSubCompOcc.Suppessed = False Then
```

```
        If Left(oSubCompOcc.Name,Len(CurrentRepName)) = CurrentRepName
```

```
            ActualSubOccCounter = ActualSubOccCounter+1
```

```
        End If
```

```
    End If
```

```
        For Each oSubSubCompOcc In
```

```
oSubCompOcc.SubOccurrences
```

```
            If oSubSubCompOcc.Suppessed = False Then
```

```
                If Left(oSubSubCompOcc.Name,Len(CurrentRepName)) = CurrentRepName
```

```
                    ActualSubSubOccCounter = ActualSubSubOccCounter+1
```

```
                End If
```

```
            End If
```

```
        Next
```

```
    Next
```

```
End If
```

```
Next
```

```
'Compare actual component counts with the current view rep counts
```

```
TestEqual =
```

```
String.Compare(CStr(ActualOccCounter),CStr(Left(CurrentOccCounter,Len(CStr(ActualOccCounter))))  
,True)
```

```
'Compare actual subcomponent counts with the current view rep counts
```

```
SubTestEqual =  
String.Compare(CStr(ActualSubOccCounter),CStr(Left(CurrentSubOccCounter,Len(CStr(ActualSubOcc  
Counter))))),True)
```

```
'Compare actual sub subcomponent counts with the current view rep counts
```

```
SubSubTestEqual =  
String.Compare(CStr(ActualSubSubOccCounter),CStr(Left(CurrentSubSubOccCounter,Len(CStr(Actual  
SubSubOccCounter))))),True)
```

```
'if counts don't match then inform user and allow update rules to be run
```

```
If TestEqual <> 0 Or SubTestEqual <> 0 Or SubSubTestEqual <> 0 Then
```

```
UpdateNow = MessageBox.Show("The View Rep Containing Parts Beginning With '" &  
CurrentRepName & "' Needs Updating" _
```

```
& vbCrLf & vbCrLf & "Update All 'Begins' View Reps Now?","iLogic View  
Representations",MessageBoxButtons.YesNo)
```

```
    If UpdateNow = vbYes Then
```

```
        iLogicVb.RunRule("Update Begins")
```

```
    End If
```

```
Return
```

```
End If
```

```
End If
```

```
Next
```

Update 'Begins' iLogic Rule

```
'Start of iLogic code
```

```
'Note to self – subscribe to Luke Davenport's blog and follow him on
```

```
'twitter @LukeCadline
```

```

'define current document

Dim openDoc As Document

openDoc = ThisDoc.Document

Dim oAsmCompDef As AssemblyComponentDefinition

oAsmCompDef = openDoc.ComponentDefinition

Dim oViewRep As DesignViewRepresentation

Try

'Activate a writeable View Rep (master view rep is not writeable)

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").activate

Catch

'Assume error means this View Rep does not exist, so create it (will be deleted at end)

oViewRep = oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Add("WIP View
Rep")

End Try

i = 0

'Find next view rep that needs updating

For Each oViewRep In oAsmCompDef.RepresentationsManager.DesignViewRepresentations

    If oViewRep.Name.Contains("Begins") Then

        oViewRep.Activate

        oViewRep.Locked = False

        i = i+1

        'Create string for current contents of view rep (this is located inside the view rep name so needs to
be pulled out)

        CurrentRepName = Mid(oViewRep.Name,9,Len(oViewRep.Name)-10-((Len(oViewRep.Name))-
((oViewRep.Name.IndexOf("("))))))

```

oOccCounter = 0

oSubOccCounter = 0

oSubSubOccCounter = 0

'look at all of the components in the assembly

Dim oCompDef As Inventor.ComponentDefinition = openDoc.ComponentDefinition

'define the first level components collection

Dim oCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubCompOcc As Inventor.ComponentOccurrence

'define the next level components collection

Dim oSubSubCompOcc As Inventor.ComponentOccurrence

'Turn off the visibility of parts in the top level assembly that don't contain the current view rep name

For Each oCompOcc in oCompDef.Occurrences

 If oCompOcc.Suppresseed = False Then

 If Left(oCompOcc.Name,Len(CurrentRepName)) = CurrentRepName Then

 oCompOcc.Visible = True

 oOccCounter = oOccCounter+1

 Else

 oCompOcc.Visible = False

 End If

 'Turn off the visibility of parts in the next level assembly that don't contain the current view rep name

 For Each oSubCompOcc In oCompOcc.SubOccurrences

 If oSubCompOcc.Suppresseed = False Then

 If Left(oSubCompOcc.Name,Len(CurrentRepName)) = CurrentRepName Then

 oSubCompOcc.Visible = True

```

oSubOccCounter = oSubOccCounter+1

Else

oSubCompOcc.Visible = False

End If

    End If

    For Each oSubSubCompOcc In oSubCompOcc.SubOccurrences

        If oSubSubCompOcc.Suppresse = False Then

            If Left(oSubSubCompOcc.Name,Len(CurrentRepName)) =
CurrentRepName Then

                oSubSubCompOcc.Visible = True

                'Increment counter for second level components

                oSubSubOccCounter = oSubSubOccCounter+1

            Else

                oSubSubCompOcc.Visible = False

            End If

        End If

    Next

Next

    End If

Next

Next

'Rename View Rep to include component counts

oViewRep.Name = ("Begins '" & CurrentRepName & "' (Qty: " & oOccCounter & " Top Level, " &
oSubOccCounter & " Sub Comps, " _
& oSubSubOccCounter & " Sub Sub Comps)")

'lock the new view rep

oViewRep.Locked = True

End If

```

Next

Try

'Delete WIP view rep

oAsmCompDef.RepresentationsManager.DesignViewRepresentations.Item("WIP View Rep").delete

Catch

End Try

'Return view to Home view

ThisApplication.CommandManager.ControlDefinitions.Item("AppViewCubeHomeCmd").Execute

If i=1 Then

MessageBox.Show(i & " View Rep Has Been Updated", "Success!")

Else If i>1 Then

MessageBox.Show(i & " View Reps Have Been Updated", "Success!")

Else

MessageBox.Show("No 'Begins' View Reps Exist" & vbCrLf & vbCrLf & "Please Create View Reps First!",
"Failed")

End If