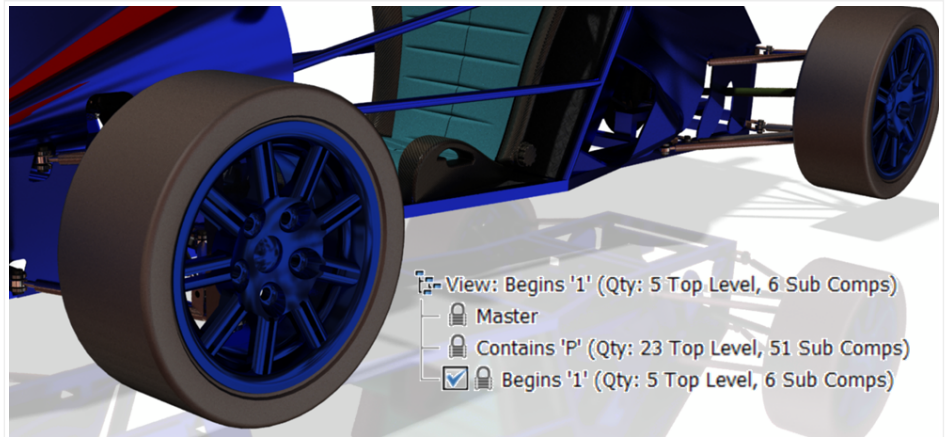


Inventor 2014 - View Representations with iLogic – Part 2 – Update

by Luke Davenport



Welcome back! This is Part 2 of my 3 part Blog on iLogic View Representations!

Question: Can you use iLogic to automatically create and update view reps based on a particular Part Number (or section of a Part Number) you specify?

Answer: Why yes, yes you can.

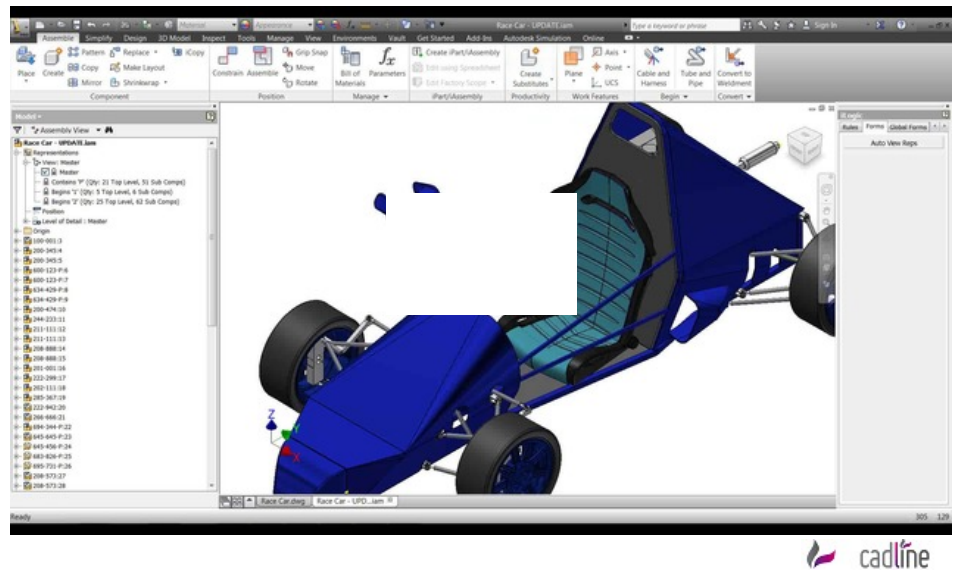
Part 2 – Updating the View Reps Automatically

I've pasted 2 iLogic rules below again. When either rule is pasted into an assembly and run, it will look for any view representations that have been automatically created (see above picture for the naming format it looks for) and updates all the visibilities within them in turn (as many view reps as it finds). Again – we've got 2 rules instead of 1 because you may have created the view reps using the 'Create Contains' rule (in which case use the 'Update Contains' rule), or the 'Create Begins' rule (in which case use the 'Update Begins' rule). – *Watch the video from my last blog if this doesn't make sense!* Either of these methods will work totally fine in isolation. Otherwise the iLogic rules are identical.

Here's what the 'Update Contains' rule will do when it is run (in sequence);

- 1) Find any existing view reps with a name that starts with 'Contains'
- 2) Update the visibilities within these in turn – also update the quantities in view rep title.
- 3) Inform user how many view reps have been updated.

As always I've recorded a wonderfully helpful video



In part 3 of this blog I'll answer this question:

Can we automatically (AND UNOBTRUSIVELY!) check whether view rep updates are required and give the user a choice to update them or not?

Enjoy!

'iLogic code starts here;

'Update Contains' Rule;

'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

'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

'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.Suppressed = 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.Suppressed = False **Then**

```

        If oSubCompOcc.Name.Contains(CurrentRepName) Then

            oSubCompOcc.Visible = True

            oSubOccCounter = oSubOccCounter+1

        Else

            oSubCompOcc.Visible = False

        End If

    End If

Next

End If

Next

'Rename View Rep to include component counts
oViewRep.Name = ("Contains '" & CurrentRepName & "' (Qty: " & oOccCounter & _
" Top Level, " & oSubOccCounter & " 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" & vbCrLf & vbCrLf & "Please Create View Reps First!", "Failed")

End If

```

Start of 'Update Begins' Rule;

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

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

```
'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
```

```
'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
```

```
'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.Suppressed = 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.Suppressed = False Then
```

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

```
        oSubCompOcc.Visible = True
```

```
        oSubOccCounter = oSubOccCounter+1
```

```
Else
```

```
oSubCompOcc.Visible = False
```

```
End If
```

```
End If
```

```
Next
```

```
End If
```

```
Next
```

```
'Rename View Rep to include component counts
```

```
oViewRep.Name = ("Begins '" & CurrentRepName & "' (Qty: " & oOccCounter & " Top Level, " & oSubOccCounter & " 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("AppViewCubeHom  
eCmd").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

Was this article helpful?
0 out of 0 found this helpful



Have more questions? [Submit a request](#)

Comments

Recently viewed articles

Inventor 2014 - View Representations
with iLogic – Part 1 – Create

Luke Davenport

Related articles

Inventor 2014 - View Representations
with iLogic – Part 1 – Create

Inventor 2014 - View Representations
with iLogic – Part 3 – Check

Autodesk Inventor 2012 - Link View
and Positional Representations using
iLogic

Autodesk Inventor 2015 – Place
Component With iLogic

Inventor 2015 iLogic – Accurately
Position Drawing Views
