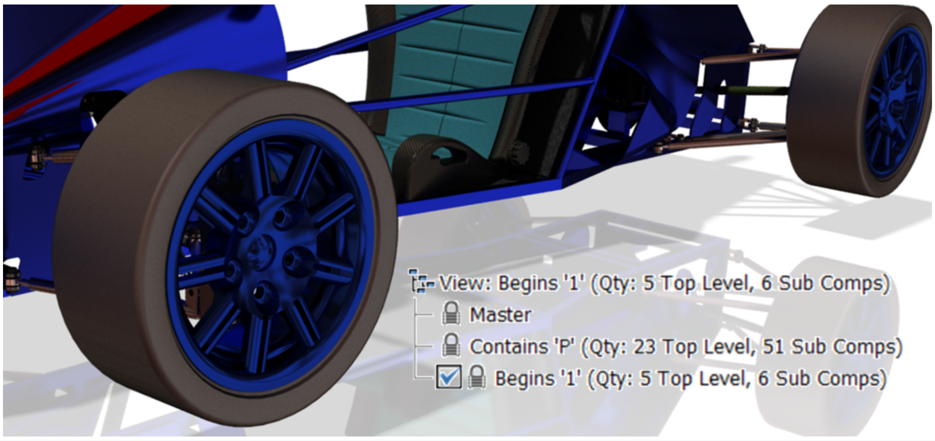
Sign in
Contact us
Log a support call

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

by Luke Davenport


View: Begins '1' (Qty: 5 Top Level, 6 Sub Comps)
Master
Contains 'P' (Qty: 23 Top Level, 51 Sub Comps)
☑ Begins '1' (Qty: 5 Top Level, 6 Sub Comps)

Welcome back! This is the final and 3rd part of my 3 Blogs on iLogic View Representations!

Part 3 – Automatically Check if View Rep Updates are Required.

See Previous Blogs to get up to speed with this. So now we've got our auto-created view reps and we can run the Update rules to update them whenever we want. However it'd be nice if the user could 'create and forget' the view reps and be prompted when saving/closing the document to update them *only if required.* Hence Part 3 here.

I've pasted the final 2 iLogic rules below. Either (or both) rules should be pasted into an assembly and set to the required event trigger (eg 'before close document'). It will then run before the document is closed but will only prompt the user to run the 'Update' rule if the update is actually required. How refreshing.

Once again – we've got 2 rules instead of 1. You may have created the view reps using the 'Create Contains' rule (in which case use the 'Check for Update - Contains' rule), or the 'Create Begins' rule (in which case use the 'Check for Update - Begins' rule). – *Watch the videos if this doesn't make sense!* Either of these methods will work totally fine in isolation. Otherwise the rules are identical.

Here's what the 'Check for Update - Contains' rule will do when it is run (using an event trigger);

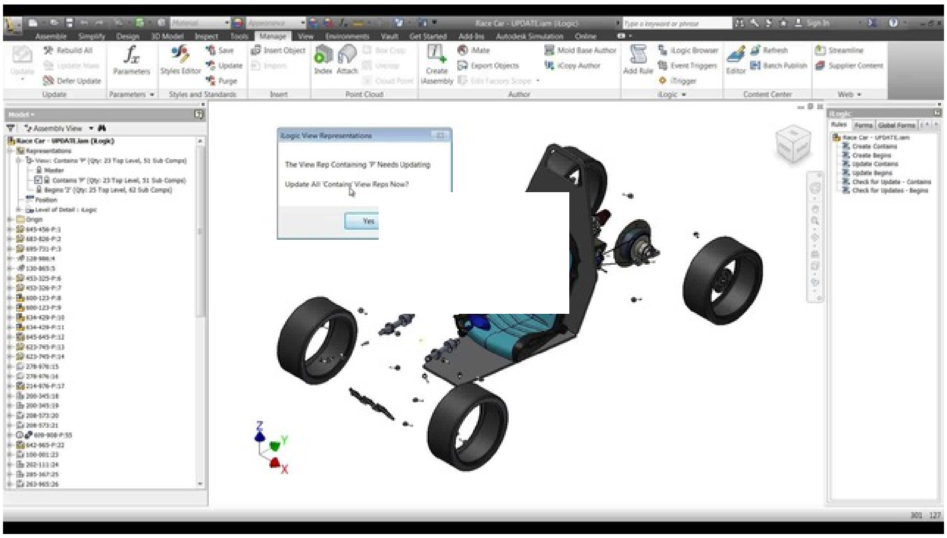
1) Find any existing view reps with a name that starts with 'Contains'

2) Run a count to see if the visibility quantities displayed in the view rep title are up-to-date.

3) Stop as soon as an out-of-date count is found and ask the user whether they want to update the rules (in this case only the 'Contains' rules).

4) If they click 'Yes' then run Update rule. If they click 'No' then exit.

And yes of course there is a wonderfully informative video



iLogic View Representations

The View Rep Containing 'F' Needs Updating

Update All 'Contains' View Reps Now?

Hope you get some use out of this. Let me know if you have other ideas for some iLogic code you’d like to see!

Luke

```
‘iLogic code starts here;


‘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

  '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))

  '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
  '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 oCompOcc.Name.Contains(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.Suppressed = False Then
            If oSubCompOcc.Name.Contains(CurrentRepName) Then
            ActualSubOccCounter = ActualSubOccCounter+1
            End If
          End If
        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(ActualSubOccCounter)))),True)

'if counts don't match then inform user and allow update rules to be run
If TestEqual <> 0 Or SubTestEqual <> 0 Then
UpdateNow = MessageBox.Show("The View Rep Containing '" & CurrentRepName & "' Needs Updating" _
& vbLf & vbLf & "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


'-----------------------------------------------------------------------------------------------------------------------------------------
```

Start of 'Check for 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

'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

    '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))

    '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
    '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.Suppressed = False Then
                        If Left(oSubCompOcc.Name,Len(CurrentRepName)) = CurrentRepName
                        ActualSubOccCounter = ActualSubOccCounter+1
                        End If
```

```
          End If
        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(ActualSubOccCounter)))),True)

'if counts don't match then inform user and allow update rules to be run
If TestEqual <> 0 Or SubTestEqual <> 0 Then
UpdateNow = MessageBox.Show("The View Rep Containing Parts Beginning With '" & CurrentRepName & "' Needs Updating" _
& vbLf & vbLf & "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
```

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 2 – Update
- 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 2 – Update
- Autodesk Inventor 2012 - Link View

and Positional Representations using iLogic

Inventor 2014 - iLogic – Set Parameter MultiValue List

Trouble Shooting Inventor iLogic Error Messages