HI Chandra,

I was not able to get the code above to work for me. I got the following error,” The parameter is incorrect. (Exception from HRESULT: 0x80070057 (E\_INVALIDARG))” However, with the help of a coworker I was able to get one of the other codes (Rule 20) in my file working. The code has been modified to this:

**Dim** doc **=** **ThisApplication**.ActiveDocument

**Dim** entity **=** **ThisApplication**.CommandManager.Pick**(**SelectionFilterEnum.kDrawingCurveSegmentFilter, "select line"**)**

*' Set a reference to the drawing document.*

*' This assumes a drawing document is active.*

**Dim** oDrawDoc **As** DrawingDocument **=** **ThisApplication**.ActiveDocument

*' Set a reference to the active sheet.*

**Dim** oActiveSheet **As** Sheet **=** oDrawDoc.**ActiveSheet**

*' Set a reference to the drawing curve segment.*

*' This assumes that a drwaing curve is selected.*

**Dim** oDrawingCurveSegment **As** DrawingCurveSegment **=** entity

*' Set a reference to the drawing curve.*

**Dim** oDrawingCurve **As** DrawingCurve **=** oDrawingCurveSegment.Parent

*' Get the mid point of the selected curve*

*' assuming that the selection curve is linear*

**Dim** oMidPoint **As** Point2d **=** oDrawingCurve.MidPoint

*' Set a reference to the TransientGeometry object.*

**Dim** oTG **As** TransientGeometry **=** **ThisApplication**.TransientGeometry

**Dim** oLeaderPoints **As** ObjectCollection **=** **ThisApplication**.TransientObjects.CreateObjectCollection

*' Create a few leader points.*

**Call** oLeaderPoints.Add**(**oTG.CreatePoint2d**(**oMidPoint.X, oMidPoint.Y **-** **1.8))**

**Call** oLeaderPoints.Add**(**oTG.CreatePoint2d**(**oMidPoint.X, oMidPoint.Y **-** **1.8** **))**

*' Create an intent and add to the leader points collection.*

*' This is the geometry that the symbol will attach to.*

**Dim** oGeometryIntent **As** GeometryIntent **=** oActiveSheet.CreateGeometryIntent**(**oDrawingCurve**)**

**Call** oLeaderPoints.Add**(**oGeometryIntent**)**

*'Dim oNote As String*

*' oNote = oDrawDoc.ActiveSheet.DrawingNotes.LeaderNotes.Item(1).FormattedText*

*' 'MessageBox.Show( oNote, "Title")*

*'Dim sText As String*

**Dim** sText **As** **String**

*'ItemLetters = Parameter("J19708 metal item A.iam.ItemLetter")*

*'sText = oNote*

*' sText =oDrawDoc.ActiveSheet.DrawingNotes.LeaderNotes.Item(1).FormattedText*

sText **=**oDrawDoc.**ActiveSheet**.DrawingNotes.LeaderNotes.Item**(1)**.FormattedText

*'MessageBox.Show( sText, "Title")*

*'sText = "<Parameter Resolved='True' ComponentIdentifier='" & strFilename & "' Name='ItemLetter' Precision='2'>" & ItemLetters & "</Parameter>"*

**Dim** oLeaderNote **As** LeaderNote **=** oActiveSheet.DrawingNotes.LeaderNotes.Add**(**oLeaderPoints, sText**)**

*'*

*' ' Insert a node.*

*' Dim oFirstNode As LeaderNode = oLeaderNote.Leader.RootNode.ChildNodes.Item(1)*

*'*

*' Dim oSecondNode As LeaderNode = oFirstNode.ChildNodes.Item(1)*

*'*

*' Call oFirstNode.InsertNode(oSecondNode, oTG.CreatePoint2d(oMidPoint.X + 5, oMidPoint.Y + 5))*

*'End Sub*

*'*

*'' Get the first symbol definition*

*'Dim oSketchSymDef As SketchedSymbolDefinition = oDrawDoc.SketchedSymbolDefinitions.Item("test leader")*

*'*

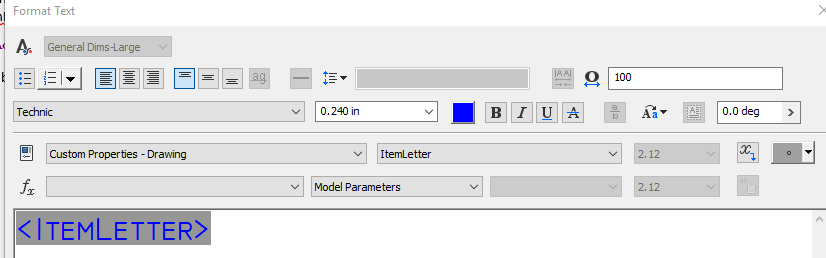
*'' Create the symbol with a leader*

*'Dim oSketchedSymbol As SketchedSymbol = oActiveSheet.SketchedSymbols.AddWithLeader(oSketchSymDef, oLeaderPoints)*

The one quirk is that I have to have a dummy leader set off to the side so that the code can refer to it in order to pull our custom property “ItemLetter”. Here is the line of code:

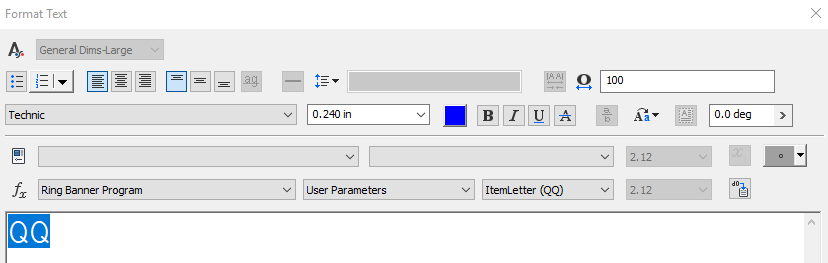
sText **=**oDrawDoc.**ActiveSheet**.DrawingNotes.LeaderNotes.Item**(1)**.FormattedText

If you open the leader box it looks like this:



This has been our best work around for this but I keep thinking there has to be a better way to do this.

The assembly we are labeling has a user parameter, “ItemLetter” as well. I was able to manually create a leader that worked really well, it would update anytime the ItemLetter was changed but I don’t know how to create the code for this. The manually created leader dialog box looks like this:



Is there a way to change this line of code so that it will insert the User Parameter from the assembly?

sText **=**oDrawDoc.**ActiveSheet**.DrawingNotes.LeaderNotes.Item**(1)**.FormattedText

That would be ideal. Then I need to figure out how to do the following two things:

1. Have this automated, as in automatically pick the parts and place the leaders. Right now I have to run it and then pick a part. We are trying to fully automate this print so the engineers can focus on larger, more complex projects.
2. I need to figure out the code so that the ItemLetter is followed by a number that increases by one with each new part. Ex QQ1, QQ2, QQ3 and so on.

Thanks for the help!