

Feature

Graphic Commands – Draw Leaders On Cells

by Mark Stefanchuk

In my last article I demonstrated how to draw and edit a right of way pie symbol. At the end of the article I provided a list of improvements and in this article I thought I might tackle one of these enhancements, leader line placement.

For variety I've decided to use a different symbol this time, a right of way structure symbol. The command places a triangle with a letter at its center. This placement command is identical to the pie symbol and only differs in the type and number of elements placed. A complete example is available on www.markstefancuk.com along with the code for my previous article Graphic Command Design.

Leader Segments

Just as we had with the pie symbol the structure triangle command acts and feels like the Place Cell command. What if we could place a cell but make it act and feel like MicroStation's Place Note command? In other words place the cell with a leader connected to it pointing to the feature represented by the symbol.

Some may ask, why not simply use a graphic group? Well, there is no right or wrong way to design a command. There are however, practices that are better than others. Graphic groups have historically been problematic. Managing the graphic group numbers in older products would often become complex leading to bugs within the programs. For this reason many of us have moved away from graphic groups, choosing cells instead.

Further, cells provide a means of programmatically manipulating the elements as a single entity, while graphically grouped elements must be transformed and manipulated individually making the commands more complicated. For example, consider a transformation where we moved the cell created at the design file origin to the cursor location. A single line of code is required.

```
\ move structure to point and apply active settings  
oCell.Transform Transform3dFromXYZ(Point.X, Point.Y, Point.Z)
```

Code 1 Transform The Cell Only

Using graphic groups, each element must be transformed.

```
oShape.Transform Transform3dFromXYZ(Point.X, Point.Y, Point.Z)  
oText.Transform Transform3dFromXYZ(Point.X, Point.Y, Point.Z)  
oLine.Transform Transform3dFromXYZ(Point.X, Point.Y, Point.Z)
```

Code 2 Transform All Elements That Compose The Symbol

As you might imagine a symbol containing more elements would become exceedingly more complicated to code especially if there are also multiple transformations applied.

So, now how do we draw leaders and at the same time include them as elements of our cell? We'll start first by considering just one leader segment without a terminator.

Think of the command similar to place line where two data points are required to define the element. It just happens that at the dynamic point is a symbol, a triangle with a letter in it.

Step 1 Declare Local Global Variables

The first step is to declare a local global variable for counting the points. And for adding the element to the design file you will need to save the first point. Let's call these `PointCount` (an Integer) and `p1` (a `Point3d`) respectively.

```
Option Explicit
Implements IPrimitiveCommandEvents

Dim PointCount As Integer
Dim p1 As Point3d
Dim MyKeyIn As String
```

Code 3 Global Variables For PlaceTriangle

Step 2 Initialize Point Count

In the Start handler we need to initialize `PointCount` so that each time the command is restarted it has a value of 0.

```
Private Sub IPrimitiveCommandEvents_Start()
    MyKeyIn = "A"
    PointCount = 0
    ShowCommand "Place Structure"
    ShowPrompt = "Enter Structure Origin"
    CommandState.StartDynamics
End Sub
```

Code 4 Initialize PointCount = 0

Step 3 Add A Parameter For the Start Point Of The Leader Line

The draw routine must handle a parameter for the start point of the line. We will use `Point` to define the end point of the leader.

```
Private Sub DrawStructure (pt1 As Point3d, Point As Point3d, _
    view as view, DrawMode As MsdDrawingMode)
```

Code 5 Accepting A Start Point Parameter pt1

Step 4 Draw The Leader And Reverse The Transformations

The parameter `pt1` is only used when `PointCount > 0`. And when this is the case a new line element needs to be created for the cell. Also, be certain that the number of elements contained in `oElts` is large enough, in this case we need 3. VBA will raise an error when the you try to access an array index that doesn't exist.

```
` add the leader segment
If PointCount > 0 Then
    Set oElts(2) = CreateLineElement2(Nothing, pt1, Point)
    ApplyReverseTransform oElts(2), Point, view
End If
```

Code 6 Creating The Leader Segment From pt1 To Point

Notice that a reverse transform has been applied. You must apply an opposite transform because the leader lines are not being drawn about the origin where the cell is being drawn. Code 7 shows how to handle reverse transformations.

```

Public Function ApplyReverseTransform (oElt As Element, _
    Point As Point3d, view As view) As Boolean
    ApplyReverseTransform = False
    On Error GoTo ApplyReverseTransformError

    Dim Origin As Point3d, tmppt as Point3d
    Origin = Point3dZero
    tmppt.X = 0 - Point.X
    tmppt.Y = 0 - Point.Y
    tmppt.Z = 0 - Point.Z

    ` move to point
    oElt.Transform = Transform3dFromXYZ(tmppt.X, tmppt.Y, tmppt.Z)

    ` active scale
    Dim ScaleMatrix As Matrix3d
    ScaleMatrix = Matrix3dFromScale(1 / ActiveSettings.Scale.X)
    oElt.Transform Transform3dFromMatrix3dAndFixedPoint3d (ScaleMatrix, origin)

    ` rotate by active angle
    Dim rMatrix As Matrix3d
    Dim axis As Point3d
    axis.X = 0: axis.Y = 0: axis.Z = 1
    rMatrix = Matrix3dFromVectorAndRotationAngle(axis, 0 - ActiveSettings.angle)
    oElt.Transform Transform3dFromMatrix3dAndFixedPoint3d(rMatrix, origin)

    ` view rotation
    Dim rAngle As Double
    Dim rAngle As Double
    If Matrix3dIsXYRotation(view.Rotation, rAngle) Then
        rMatrix = Matrix3dFromVectorAndRotationAngle(axis, rAngle)
        oElt.Transform Transform3dFromMatrix3dAndFixedPoint3d(rMatrix, origin)
    End If
    ApplyReverseTransform = True
    Exit Function
ApplyReverseTransformError:
Debug.Print "Error: Reverse Transform"
End Function

```

Code 7 Reverse Transform Function

Step 5 Handle Leader Line Points And Final Call To Draw Routine

In the data point handler we have two distinct operations. When `PointCount` is 0 we save the first point in `p1` so we can use it to draw the leader, and increment `PointCount` so that the next time a data point is entered the command can draw the symbol and restart the command.

```

` DATAPOINT HANDLER
Private Sub IPrimitiveCommandEvents_DataPoint(Point As Point3d, ByVal view As view)
    If PointCount = 0 Then
        PointCount = 1
        p1 = Point
    Else
        DrawStructure p1, Point, view, msdDrawingModeNormal
        CommandState.StartPrimitive Me, True
    End If
End Sub

```

Code 8 PointCount Determines When To Save p1 And When To Draw

Step 6 Draw The Leader Line Dynamically

The dynamics handler just calls the draw routine. It doesn't matter that p1 may not be defined because the draw routine doesn't use it until the datapoint handler sets PointCount to 1.

```
` DYNAMIC HANDLER
Private Sub IPrimitiveCommandEvents_Dynamics(Point As Point3d, _
    ByVal view As view, ByVal DrawMode As MsdDrawingMode)
    DrawStructure p1, Point, view, DrawMode
End Sub
```

Code 9 PointCount Determines When To Draw Leader

But the symbol with the leader isn't quite correct. You will notice that the leader element draws to the center of the symbol. It would be much better if the leader line were trimmed back to the edge of the shape.

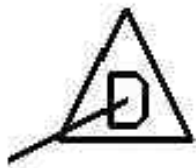


Figure 1 Leader Draws To Center But Should Draw To Triangle Edge

Step 7 Find The Intersection Of The Shape And The Line

Making the leader line stop at the edge of the triangle requires finding the intersection of the triangle shape and the line and then drawing the line from p1 to the intersection point. The following code is part of the draw routine DrawStructure.

```
` add the leader segment
If PointCount > 0 Then
    Dim ipt() As Point3d

    ` move shape to where the line is being drawn
    ComponentTransform oElts(0), Point, view

    ` draw line
    Set oElts(2) = CreateLineElement2(Nothing, pt1, Point)

    ` find the intersection of the line and the shape
    ipt() = oElts(0).AsShapeElement_
        .GetIntersectionPoints(oElts(2).AsLineElement, Matrix3dIdentity)

    ` put the shape back
    ApplyReverseTransform oElts(0), Point, view

    ` test for intersection
    If Ubound(ipt) > -1 Then
        ` redraw the line
        Set oElts(2) = CreateLineElement2(Nothing, pt1, ipt(0))
    End If

    ApplyReverseTransform oElts(2), Point, view
End If
```

Code 10 Leader Draws To Center But Should Draw To Triangle Edge

Notice that we moved the shape to the location of the leader segment and then back again. And we also implemented a test to make sure that the intersection exists. Attempting to draw a line from a point to an undefined point will raise an error.

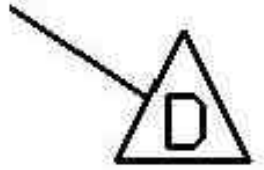


Figure 2 Leader Draws To Center But Should Draw To Triangle Edge

Step 8 Two Segment Leaders Need Another Vertex

To handle a two segment leader you must declare a second global point, `p2`.

```
Option Explicit
Implements IprimitiveCommandEvents

Dim PointCount As Integer
Dim p1 As Point3d
Dim p2 As Point3d
Dim MyKeyIn As String
```

Code 11 Declare New Point3d p2

You will need to add a new parameter to your draw routine `pt2`. And don't forget to increment the element array `oElts()`.

```
Private Sub DrawStructure (pt1 As Point3d, pt2 As Point3d, _
    Point As Point3d, view As view, DrawMode As MsdDrawingMode)
```

Code 12: Add New Point3d Parameter pt2

Why not just use `p1`, and `p2` in the draw routine? Why pass a new parameter? I do this because I may call the draw routine from another command. If I do that then I will move the routine out of the class and into a standard module. And I will change the routine from `Private` to `Public`. This makes the routine visible throughout the project.

The global variables are local and are easy to maintain within the class context. Moving them into a Module however, will make them harder to maintain, and it's likely that I will forget they have been used and try to re-use them someplace else in my code, and that's not good programming practice.

Tips

Enable/Disable Edit Handles Key-in

Is there a key-in or other easier way to Enable/Disable Edit Handles?

Try

```
mdl load calculat;calc userPrefsP->addFlags.hiliteSelectionSet=0 (for Off)
mdl load calculat;calc userPrefsP->addFlags.hiliteSelectionSet=1 (for On)
```

CAD

Tips - v8 Documentation

Guide to Using DWG and Other Formats: Enabling and Disabling Workmode Capabilities

<http://docs.bentley.com/en/MicroStation/ustnhelp1047.html>

CAD

Code 8 demonstrates how to draw a leader segment to the edge of the triangle. Moving this into it's own method we would have something like this.

```
Private Sub DrawSegment(e0 As element, e2 As element, _
    point As Point3d, view As view, pt1 As Point3d)
    Dim ipts() As Point3d
    ComponentTransform e0, point, view
    Set e2 = CreateLineElement2(Nothing, pt1, point)
    ipts() = e0.AsShapeElement _
        .GetIntersectionPoints(e2, Matrix3dIdentity)
    ApplyReverseTransform e0, point, view
    If Ubound (ipts) > -1 Then
        Set e2 = CreateLineElement2(Nothing, pt1, ipts(0))
    End if
    ApplyReverseTransform e2, point, view
End Sub
```

Code 13 Consolidate Leader Segment Code Into Common Sub Routine

In the draw routine we want to watch for the first datapoint which indicates we are drawing the first segment of the leader. Then upon receiving a second datapoint we want to draw the second segment. Notice that the first segment is drawn the same way as before as long as `PointCount < 2`. But when `PointCount` is 2 the first segment is drawn between the `pt1` and `pt2`. It is the second segment that must find the intersection of the shape and the line.

```
` draw leader lines
If PointCount = 1 Then
    DrawSegment oElts(0), oElts(2), point, view, pt1
ElseIf
    Set oElts(2) = CreateLineElement2(Nothing, pt1, pt2)
    ApplyReverseTransform oElts(2), point, view
    DrawSegment oElts(0), oElts(2), point, view, pt1
End If
```

Code 14 Handling Leader Lines In Draw Routine

Now, the datapoint handler must also be extended to watch for the first and second points. Dynamics `p1` and `p2` are not defined because the draw routine doesn't use them until they have been defined. The datapoint handler defines these as each datapoint is entered. So when the first datapoint is entered by the user `p1` equals `Point`, the second datapoint means `p2` is `Point`. The next data point indicates the user wants to finalize placement and draw the symbol.

```
` DATAPOINT HANDLER
Private Sub IPrimitiveCommandEvents_DataPoint(Point As Point3d, ByVal view As view)
    If PointCount = 0 Then
        p1 = Point
        PointCount = PointCount + 1
    ElseIf PointCount = 1 Then
        p2 = Point
        PointCount = PointCount + 1
    End if
End Sub

` DYNAMICS HANDLER
Private Sub IPrimitiveCommandEvents_Dynamics(Point As Point3d, _
    ByVal view As view, ByVal DrawMode As MsdDrawingMode)
    DrawStructure p1, p2, Point, view, DrawMode
End Sub
```

Code 15 DataPoint and Dynamics Extended to Watch For PointCount

Step 9 Adding A Terminator

Draw a circle at pt1. That means that the element array oElts must be increased one more time to oElts(5). The circle can be defined at pt1 for PointCount > 0.

```
` Draw Terminator
If PointCount = 1 Then
  Set oElts(3) = CreateEllipseElement2(Nothing, pt1, _
    0.125 * th, 0.125 * th, Matrix3dIdentity, msdFillModeFilled)
  ApplyReverseTransformation oElts(3), point view
ElseIf PointCount > 1 Then
  Set oElts(4) = CreateEllipseElement2(Nothing, pt1, _
    0.125 * th, 0.125 * th, Matrix3dIdentity, msdFillModeFilled)
  ApplyReverseTransformation oElts(4), point view
End if
```

Code 16 Draw Terminator Circle

The ellipse element draws a circle when the primary and secondary radii are equal. Using text height and a scale factor the terminator has a rule for sizing the circle so that it is proportional to the symbol and will grow/shrink when the size of the symbol is manipulated. Notice too that the oElts must be sequential and cannot skip an index. If we were to try and use element index 4 when PointCount is 1 the leader segment will not draw.

About The Author

Mark Stefanchuk is a consultant and contract programmer specializing in business and design workflow automation. He was co-founder and principle architect of cadgurus.com a web community for MicroStation professionals.

More recently Mr. Stefanchuk continues to work on projects involving MDL, VBA, and Microsoft Windows technology. Mark may be contacted by sending email to markastefanchuk@yahoo.com.

CAD

Tips - v8

Modifying Elements with Tags Attached

In V8 2004, I have loss the ability to rotate tags when rotating the elements associated to tags. In V7, with the variable MS_TAGTRANSFORM WITHBASSELEMENT defined, this was possible.

Unfortunately no, you also have to select the tag elements. This functionality will return in a future MicroStation version (Mozart).

CAD

Tips

Unable to Write to Cell Library

Somehow I've made one of my cell libraries read only. When I try to "write" a cell to it - I get the error message "unable to write to file".

In Windows explorer right click on the cell library and select properties. In the "Attributes" section, uncheck the Read-only box.

CAD

Tips - v7, v8

Redundant "Design" Key-in

In v7, a key-in "design" will bring up a dialog box called "Design Options", which contains buttons for other dialogs for Data Readout, Working Units, Active Angle, Active Scale, etc. What is the equivalent key-in for v8?

That dialog box is now gone. The Design Options dialog which was a little different from the Design Settings dialog in that it opened some dialogs that are part of Design Settings as well as the Reference dialog, the Cell maintenance dialog, etc.

You can create a simple VBA macro that opens a form and performs the same end result (which is better than the original, as you can include other options as you desire).

CAD

Subscribe to ControlAltDelete

Keep in touch with the MicroStation World
See page 29 for details

<http://www.penbrush.com>