

Graphic Commands – Changing Leaders

Even the simplest designs require changes, and for the most part MicroStation provides a comprehensive set of modification tools to assist you in making changes. Customized commands however, such as the symbols with leaders from my previous articles are not easily changed. They can be moved, copied, or mirrored like any MicroStation cell. But if we want to extend the leader or move the symbol the leader and terminator move too. That's probably not the desired result.

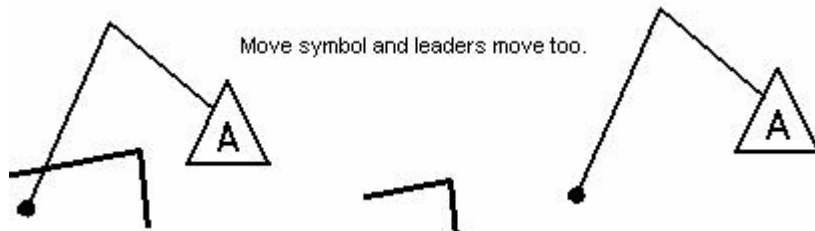


Figure 1: Move Symbol And Leaders Move

In the following pages I'll extend the discussion on graphic command design to include modifications to symbols with leaders. Our new tools will move a symbol but maintain the same anchor point allowing us to stretch the leader. I will also demonstrate how to construct commands for converting a two segment leader to a single segment, removing leaders, and adding leaders to the symbol.

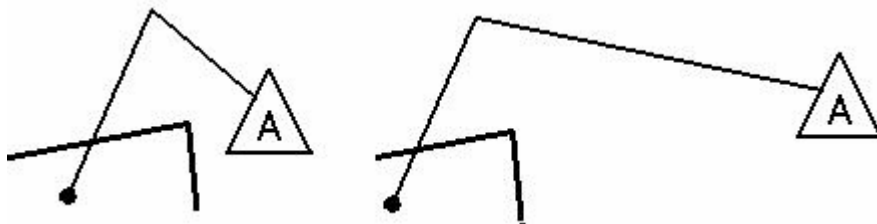


Figure 2: Move Symbol Anchored

The easy solution might be to delete and redraw. But, I have found that designers and production users don't work that way. It's more likely that the cell will be dropped to its simplest components and then modified. With custom commands this workflow is never welcome since tags, and other intelligent data will be lost. Instead include in your application commands that help your users work more intuitively.

Anchoring The Terminator

The structure symbol I introduced in my previous article may or may not have a leader and if it does have leader the leader may have one or two segments. To move the symbol each of these conditions must be handled differently.

1) No Leader – Move the symbol like MicroStation move command would move it.



Figure 3: Move Symbol

2) One Leader Segment – Leave terminator symbol fixed. Redraw segment between terminator and symbol.

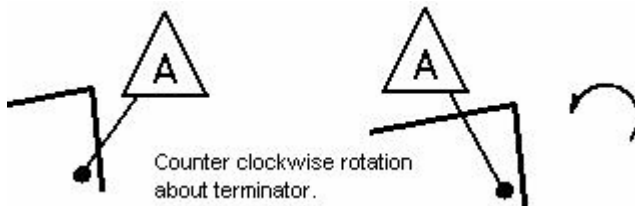


Figure 4: Anchor Terminator

3) Two Leader Segment – Leave terminator and first segment fixed. Redraw second segment from vertex to symbol.

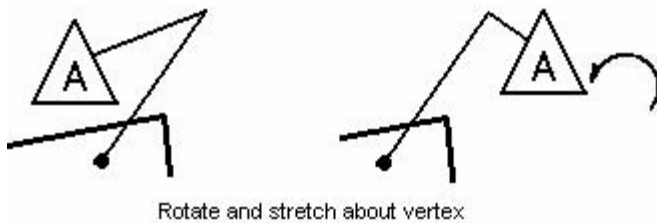


Figure 5: Anchor Terminator and First Segment

Each of these are decisions the command can make for us based on the number of elements found in the symbol. If the structure only has two elements then condition 1 applies, four elements then means condition 2, and if five elements then condition 3.

```
' count number of elements in the symbol
Set ee = c.GetSubElements
Do While ee.MoveNext
    count = count + 1
Loop
```

The variable *c* is the located cell and is passed to our dynamics function. An element enumerator is assigned the elements in the cell and then the Do While is used to count the number of elements.

When there are no segments, or count is some value other than 4 or 5, the command acts like MicroStation move.

```
' no segments so just move the symbol
Dim tpt As Point3d, ndx As Integer
tpt = Point3dSubtract(Point, c.origin)
ee.Reset
ndx = 0
Do While ee.MoveNext
    Set oElts(ndx) = ee.Current
    oElts(ndx).Transform Transform3dFromXYZ(tpt.X, tpt.Y, tpt.Z)
    ndx = ndx + 1
Loop
```

Each element in the cell is moved from the cell origin to the new cursor location.

When there are 4, or 5 elements in the cell we know that a new segment must be redrawn. By stepping through the elements of the cell a second time we can extract the information we need, the start points for each segment.

```
' when count is 4 there is only 1 segment
' when count is 5 there are two segments
If (ee.Current.IsLineElement) Then
    If linecount = 0 Then
        Set oLine1 = ee.Current
        If count = 4 Then
            ' terminator of the leader line
            startpoint = oLine1.startpoint
        ElseIf count = 5 Then
            ' vertex of the leader line
            startpoint = oLine1.EndPoint
        End If
    ElseIf linecount = 1 Then
        Set oLine2 = ee.Current
    End If
    linecount = linecount + 1
End If
```

The shape and text elements are also saved in the element array. No need to re-create these since they only need to be moved. Finding the end of the new leader segment requires finding the intersection of the shape and the line, which I covered in my last article. Look for the download containing complete source code on <http://www.markstefanchuk.com/>.

So, moving the shape, text, and then redrawing the leader line is all that needs to be done.

Removing The Vertex

The nature of this command implies that the cell you are working on has a leader with two segments. It is still important that you test for the number of elements. In this case the element count must be 5, and if not then the locate filter should stop the accept function. The accept routine extracts the components from the selected cell and saves them in an element array. Line elements are discarded and are replaced by a line drawn from the terminator to the intersection of the shape.

```
' erase the original cell - but don't delete it yet
element.AsCellElement.Redraw msdDrawingModeErase

' get the center of the ellipse and save all except lines
Set ee = element.AsCellElement.GetSubElements
Do While ee.MoveNext
  ' save shape, text, and ellipse
  If ee.Current.IsShapeElement Then
    Set oElts(0) = ee.Current
  End If

  ' save text element
  If ee.Current.IsTextElement Then
    Set oElts(1) = ee.Current
  End If

  ' element order is important the ellipse is always the last element
  If (ee.Current.Type = msdElementTypeEllipse) Then
    Set oElts(3) = ee.Current
  End If
Loop

' new line segment
Set oElts(2) = CreateLineElement2(Nothing, _
  oElts(3).AsEllipseElement.CenterPoint, _
  element.AsCellElement.origin)

' get the intersection
Dim ipt3d() As Point3d
ipt3d() = oElts(0).AsShapeElement _
  .GetIntersectionPoints(oElts(2).AsLineElement, Matrix3dIdentity)

If UBound(ipt3d) > -1 Then
  Set oElts(2) = CreateLineElement2(Nothing, _
  oElts(3).AsEllipseElement.CenterPoint, ipt3d(0))
End If
```



Figure 6: Remove Vertex

Removing Leaders

The simplest of the four modification commands, removing the leaders requires that we extract the structure elements (triangle, and text) and then re-create the cell with just these two elements. The locate filter tests for cells with more than 3 elements. No need to do anything if the symbol doesn't have a leader.

```
Public Sub RemoveAllLdrs(oCell As CellElement)
    Dim ee As ElementEnumerator, oNewCell As CellElement
    Dim oElts(2) As element

    ' erase the original cell - but don't delete it yet
    oCell.Redraw msdDrawingModeErase

    ' get the center of the ellipse and save all except lines
    Set ee = oCell.GetSubElements
    Do While ee.MoveNext
        ' save shape, text, and ellipse
        If ee.Current.IsShapeElement Then
            Set oElts(0) = ee.Current
        End If

        ' save text element
        If ee.Current.IsTextElement Then
            Set oElts(1) = ee.Current
        End If
    Loop

    ' create the new cell
    Set oNewCell = CreateCellElement1("Structure", _
        oElts, oCell.origin, False)

    ' delete original and add new cell
    ActiveModelReference.RemoveElement oCell
    ActiveModelReference.AddElement oNewCell
    oNewCell.Redraw msdDrawingModeNormal
End Sub
```



Figure 7: Remove Leaders

I put this action into a Public Sub so that I would be able to re-use it. In the next command, Add Leaders this sub is called when users select symbols that already have leaders.

Add Leaders

The Add Leader command is a compound command. I call it that because it is a locate command that calls a primitive class upon accept.

```
CommandState.StartPrimitive New classDrawLdrPlace
```

This place leader class then draws the new lines handling the placement points and recreating the cell. Since I can't pass the located cell from one class to the next I have defined a global variable called `oCellHolder` of type `CellElement`.

The remaining code is a dynamic placement of the leader segments and is code we have seen in other examples.



Figure 8: Add New Leaders

Including modification commands with your graphic commands is essential to maintaining data integrity. And while work flow between disciplines will vary, change management is always necessary. Just ask your designers if they prefer delete and redraw or extend.