```vb
Imports System.Collections.Generic

Imports Grasshopper.Kernel
Imports Rhino.Geometry
Imports Interop.gsa_8_7
Imports UsageDataCollection


Public Class SetElement
    Inherits GH_Component
    ''' <summary>
    ''' Set an element in GSA with new or modified characteristics
    ''' </summary>
    Public Sub New()
        MyBase.New("Set Element", "Set Element", _
                "Set an element in GSA with new or modified characteristics", _
                "GsaComTools", "Elements")
    End Sub

    ''' <summary>
    ''' Registers all the input parameters for this component.  SET, EL, 11, , NO_RGB, BEAM, 1, 101, 12, 22, 3
    ''' </summary>
    Protected Overrides Sub RegisterInputParams(pManager As GH_Component.GH_InputParamManager)
        pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
        pManager.AddIntegerParameter("Element Number", "Record", "Element Number.  Leave blank to add element with arbitrary element number", GH_ParamAccess.item)
        pManager.AddIntegerParameter("Element Type", "Type", "Element Type: " & vbCrLf & _
                                                                                            "1 - Bar" & vbCrLf & _
                                                                                            "2 - Beam" & vbCrLf & _
                                                                                            "3 - Spring" & vbCrLf & _
                                                                                            "4 - Mass" & vbCrLf & _
                                                                                            "5 - Quad 4" & vbCrLf & _
                                                                                            "6 - Quad 8" & vbCrLf & _
                                                                                            "7 - Triangle 3" & vbCrLf & _
                                                                                            "8 - Triangle 6" & vbCrLf & _
                                                                                            "9 - Link" & vbCrLf & _
                                                                                            "10 - Cable" & vbCrLf & _
                                                                                            "18 - Ground Spring" & vbCrLf & _
                                                                                            "19 -
```

```
                            Spacer" & vbCrLf & _
                                                                                 "20 - 
                            Strut" & vbCrLf & _
                                                                                 "21 - Tie"
                             & vbCrLf, _
                                                                                 
    GH_ParamAccess.item)
            pManager.AddIntegerParameter("Element Property", "Prop", "Element 
              Property Number", GH_ParamAccess.item)
            pManager.AddIntegerParameter("Element Group", "Group", "Element Group 
              Number", GH_ParamAccess.item)
            pManager.AddIntegerParameter("Element Ends", "Ends", "Element Ends", 
              GH_ParamAccess.list)
            pManager.AddIntegerParameter("Element Orient. Node", "OrientNode", 
              "Element Orientation Node (Optional)", GH_ParamAccess.item)
            pManager.AddNumberParameter("Element Orient. Angle", "OrientAngle", 
              "Element Orientation Angle (Optional)", GH_ParamAccess.item)
            pManager.AddBooleanParameter("Dummy", "Dummy", "Dummy (Default - 
              False)", GH_ParamAccess.item)

            pManager(1).Optional = True
            pManager(2).Optional = True
            pManager(3).Optional = True
            pManager(4).Optional = True
            pManager(5).Optional = True
            pManager(6).Optional = True
            pManager(7).Optional = True
            pManager(8).Optional = True


        End Sub

        ''' <summary>
        ''' Registers all the output parameters for this component.
        ''' </summary>
        Protected Overrides Sub RegisterOutputParams(pManager As 
          GH_Component.GH_OutputParamManager)
            pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", 
              GH_ParamAccess.item)
        End Sub

        ''' <summary>
        ''' This is the method that actually does the work.
        ''' </summary>
        ''' <param name="DA">The DA object is used to retrieve from inputs and 
          store in outputs.</param>
        Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)

            'Declare Variables
            Dim FileName As String = ""
            Dim s As Short
            Dim myDataCollector As New DataCollector
            Dim myObjectWrapper As Types.GH_ObjectWrapper = Nothing
            Dim ElNum As String
            Dim ElNumInt As Integer
            Dim ElType As Integer
            Dim ElTypeString As String
            Dim ElProp As Integer
            Dim ElGroup As Integer
            Dim ElEnds As New List(Of Integer)
            Dim ElOrientationNode As Integer
            Dim ElOrientationNodeString As String
            Dim ElOrientationAngle As Double
            Dim ElDummy As String
            Dim ElDummyBoolean As Boolean
            Dim BlankInput As Boolean = False
            Dim myCommand As String = ""
            Dim ElemRefs As New List(Of Integer)
            Dim Elements() As GsaElement = Nothing
            Dim EndString As String
            Dim CommandString As String

            'Get Input Values
            If (Not DA.GetData(0, myObjectWrapper)) Then Return
            If (Not DA.GetData(1, ElNumInt)) Then myCommand = "ADD" Else myCommand
               = "SET"
            If (Not DA.GetData(2, ElType)) Then ElType = -1
            If (Not DA.GetData(3, ElProp)) Then ElProp = Nothing
            If (Not DA.GetData(4, ElGroup)) Then ElGroup = Nothing
            If (Not DA.GetDataList(5, ElEnds)) Then ElEnds = Nothing
            If (Not DA.GetData(6, ElOrientationNode)) Then ElOrientationNode =
              Nothing
            If (Not DA.GetData(7, ElOrientationAngle)) Then ElOrientationAngle =
              Nothing
            If (Not DA.GetData(8, ElDummyBoolean)) Then ElDummy = "BLANK"


            'Get GSA Object
            Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)

            'Get existing values if in "SET" mode
            If myCommand = "SET" Then

                'Element Number
                ElNum = CStr(ElNumInt)

                'Get existing element data
                ElemRefs.Add(CInt(ElNum))
                s = GsaObj.Elements(ElemRefs.ToArray, Elements)

                'Element Type
                If ElType = -1 Then
                    If UBound(Elements) = -1 Then
                        AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Could not
                         determine element type.  Ensure element exists in GSA or
                        specify explicit value.")
                        Return
                    Else
                        ElType = Elements(0).eType
                    End If
                End If

                'Element Property
                If ElProp = Nothing Then
                    If UBound(Elements) = -1 Then
                        AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Could not determine element property.  Ensure element exists in GSA or specify explicit value.")
                        Return
                    Else
                        ElProp = Elements(0).Property
                    End If
                End If

                'Element Group
                If ElGroup = Nothing Then
                    If UBound(Elements) = -1 Then
                        AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Could not determine element group.  Ensure element exists in GSA or specify explicit value.")
                        Return
                    Else
                        ElGroup = Elements(0).Group
                    End If
                End If

                'Element Ends
                If ElEnds Is Nothing Then
                    If UBound(Elements) = -1 Then
                        AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Could not determine element ends.  Ensure element exists in GSA or specify explicit value.")
                        Return
                    Else
                        EndString = CStr(Elements(0).Topo(0))
                        For i = 1 To UBound(Elements(0).Topo)
                            EndString = EndString & ", " & Elements(0).Topo(i)
                        Next
                    End If
                Else
                    EndString = CStr(ElEnds(0))
                    For i = 1 To ElEnds.Count - 1
                        EndString = EndString & ", " & ElEnds(i)
                    Next
                End If

                'Element Orientation Node
                If ElOrientationNode = Nothing Then
                    If UBound(Elements) = -1 Then
                        ElOrientationNodeString = ""
                    Else
                        ElOrientationNodeString = CStr(Elements(0).OrientNode)
                    End If
                Else
                    ElOrientationNodeString = CStr(ElOrientationNode)
                End If

                'Element Orientation Angle
```

```vb
            If ElOrientationAngle = Nothing Then
                If UBound(Elements) = -1 Then
                    ElOrientationAngle = 0
                Else
                    ElOrientationAngle = CDbl(GsaObj.Arg(10, CStr(GsaObj.GwaCommand("GET, EL, " & ElNum))))
                End If
            End If

            'Element Dummy
            If ElDummy = "BLANK" Then
                If UBound(Elements) = -1 Then
                    ElDummy = ""
                Else
                    ElDummy = GsaObj.Arg(GsaObj.NumArg(CStr(GsaObj.GwaCommand("GET, EL, " & ElNum))) - 1, CStr(GsaObj.GwaCommand("GET, EL, " & ElNum)))
                End If
            Else
                If ElDummyBoolean = True Then ElDummy = "DUMMY" Else ElDummy = ""
            End If

            'Prep data if in "ADD" mode
        Else
            'Element Number
            ElNumInt = CInt(GsaObj.GwaCommand("HIGHEST, EL")) + 1
            ElNum = CStr(ElNumInt)

            'Element Type
            If ElType = -1 Then
                AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Element type not collected.  Specify value.")
                Return
            End If

            'Element Property
            If ElProp = Nothing Then
                AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Element property not collected.  Specify value.")
                Return
            End If

            'Element Group
            If ElGroup = Nothing Then
                AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Element group not collected.  Specify value.")
                Return
            End If

            'Element Ends
            If ElEnds Is Nothing Then
                AddRuntimeMessage(GH_RuntimeMessageLevel.Error, "Element ends not collected.  Specify value(s).")
                Return
            Else
```

```vb
                    EndString = CStr(ElEnds(0))
                    For i = 1 To ElEnds.Count - 1
                        EndString = EndString & ", " & ElEnds(i)
                    Next
                End If

                'Element Orientation Node
                If ElOrientationNode = Nothing Then
                    ElOrientationNodeString = ""
                Else
                    ElOrientationNodeString = CStr(ElOrientationNode)
                End If

                'Element Orientation Angle
                If ElOrientationAngle = Nothing Then
                    ElOrientationAngle = 0
                End If

                'Element Dummy
                If ElDummyBoolean = Nothing Then
                    ElDummy = ""
                Else
                    If ElDummyBoolean = True Then ElDummy = "DUMMY" Else ElDummy = ""
                End If

            End If


            'Build Command String
            Select Case ElType
                Case 1
                    CommandString = "SET, EL_BAR, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationNode & ", " & ElOrientationAngle & ", " & ElDummy
                Case 2
                    CommandString = "SET, EL_BEAM, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationNode & ", " & ElOrientationAngle & ", " & ElDummy
                Case 3
                    CommandString = "SET, EL_SPRING, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationNode & ", " & ElOrientationAngle & ", " & ElDummy
                Case 4
                    CommandString = "SET, EL_MASS, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElDummy
                Case 5
                    CommandString = "SET, EL_QUAD4, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationAngle & ", " & ElDummy
                Case 6
                    CommandString = "SET, EL_QUAD8, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationAngle & ", " & ElDummy
                Case 7
                    CommandString = "SET, EL_TRI3, " & ElNum & ", " & ElProp & ",
                    " & ElGroup & ", " & EndString & ", " & ElOrientationAngle & ", " & ElDummy
                Case 8
                    CommandString = "SET, EL_TRI6, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationAngle & ", " & ElDummy
                Case 9
                    CommandString = "SET, EL_LINK, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElDummy
                Case 10
                    CommandString = "SET, EL_CABLE, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElDummy
                Case 18
                    CommandString = "SET, EL_GROUND, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElDummy
                Case 19
                    CommandString = "SET, EL_SPACER, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElDummy
                Case 20
                    CommandString = "SET, EL_STRUT, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationNode & ", " & ElOrientationAngle & ", " & ElDummy
                Case 21
                    CommandString = "SET, EL_TIE, " & ElNum & ", " & ElProp & ", " & ElGroup & ", " & EndString & ", " & ElOrientationNode & ", " & ElOrientationAngle & ", " & ElDummy
            End Select

            'Execute GwaCommand
            GsaObj.GwaCommand(CommandString)

            ' Assign the data to the output parameter.
            DA.SetData(0, GsaObj)

            'Send Usage Data
            myDataCollector.UsageStatistics("SetElement")
        End Sub

        ''' <summary>
        ''' Provides an Icon for every component that will be visible in the User Interface.
        ''' Icons need to be 24x24 pixels.
        ''' </summary>
        Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
            Get
                'You can add image files to your project resources and access them like this:
                ' return Resources.IconForThisComponent;
                Return My.Resources.Element_Set_Add
            End Get
        End Property

        ''' <summary>
        ''' Gets the unique ID for this component. Do not change this ID after release.
        ''' </summary>
        Public Overrides ReadOnly Property ComponentGuid() As Guid
            Get
                Return New Guid("{ccf86cd9-e9e1-43c3-a3d0-b4500ed2979a}")
            End Get
        End Property
    End Class
```