```
Imports System.Collections.Generic

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


Public Class GwaCommand
    Inherits GH_Component
    ''' <summary>
    ''' Initializes a new instance of the GwaGet class.
    ''' </summary>
    Public Sub New()
        MyBase.New("GwaCommand", "GwaCommand", _
                "Perform GwaCommand", _
                "GsaComTools", "Misc")
    End Sub

    ''' <summary>
    ''' Registers all the input parameters for this component.
    ''' </summary>
    Protected Overrides Sub RegisterInputParams(pManager As GH_Component.GH_InputParamManager)
pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
        pManager.AddTextParameter("Command", "Command", "Command", GH_ParamAccess.item)
        pManager.AddTextParameter("Keyword", "Keyword", "Keyword", GH_ParamAccess.item)
        pManager.AddTextParameter("Data", "Data", "Data (Optional)", GH_ParamAccess.list)
        pManager.AddBooleanParameter("Save", "Save", "Save file after command? (Default is false)", GH_ParamAccess.item)

        pManager(3).Optional = True
        pManager(4).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)
        pManager.AddTextParameter("Output", "Output", "Output", GH_ParamAccess.list)
    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 Command As String
        Dim Keyword As String
        Dim Data As New List(Of String)
        Dim Elements() As GsaElement
        Dim ElementProperties As New List(Of Integer)
        Dim GwaInput As String
        Dim GwaOutput, GwaOutputItem As String
        Dim nArg As Integer
        Dim Output As New Grasshopper.DataTree(Of Object)
        Dim booSave As Boolean
        Dim i, j, ItemCount As Integer
        Dim myDataCollector As New DataCollector
        Dim myObjectWrapper As Types.GH_ObjectWrapper

        'Import GSA File Name
        If (Not DA.GetData(0, myObjectWrapper)) Then Return
        If (Not DA.GetData(1, Command)) Then Return
        If (Not DA.GetData(2, Keyword)) Then Return
        If (Not DA.GetDataList(3, Data)) Then Data = Nothing
        If (Not DA.GetData(4, booSave)) Then booSave = False

        Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)

        'Run GwaCommand
        If Data Is Nothing Then
            GwaInput = Command & "," & Keyword
            GwaOutput = GsaObj.GwaCommand(GwaInput)
            nArg = GsaObj.NumArg(GwaOutput)
            For i = 0 To nArg - 1
                Output.AddRange(New List(Of Object), New GH_Path(i))
                Output.Branches.Item(i).Add(GsaObj.Arg(i, GwaOutput))
            Next
        Else
            ItemCount = 0
            For Each item In Data
                GwaInput = Command & "," & Keyword & "," & item
                GwaOutput = GsaObj.GwaCommand(GwaInput)
                nArg = GsaObj.NumArg(GwaOutput)
                If nArg > Output.Branches.Count Then
                    For i = Output.Branches.Count To nArg
                        Output.AddRange(New List(Of Object), New GH_Path(i))
                        For j = 0 To ItemCount - 1
                            Output.Branches.Item(i).Add(Nothing)
                        Next
                    Next
                End If
                For i = 0 To Output.Branches.Count - 1
                    If i < nArg And GwaOutput <> "" Then
                        GwaOutputItem = GsaObj.Arg(i, GwaOutput)
                        If GwaOutputItem = "" Then Output.Branches.Item(i).Add(Nothing) Else Output.Branches.Item(i).Add(GsaObj.Arg(i, GwaOutput))
                    Else
                        Output.Branches.Item(i).Add(Nothing)
                    End If
                Next
                ItemCount += 1
            Next
        End If

        ' Assign the data to the output parameter.
        DA.SetData(0, GsaObj)
        DA.SetDataTree(1, Output)

        'Send Usage Data
        myDataCollector.UsageStatistics("GwaCommand")

    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.Misc_GwaCommand
        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("{3cecf442-2811-4c13-a484-81c389f99cc6}")
        End Get
    End Property
End Class
```