

```

1 Imports System.Collections.Generic
2
3 Imports Grasshopper.Kernel
4 Imports Rhino.Geometry
5 Imports Interop.gsa_8_7
6 Imports UsageDataCollection
7
8 Public Class GetElementGroups
9     Inherits GH_Component
10    ''' <summary>
11    ''' Initializes a new instance of the GetElementGroups class.
12    ''' </summary>
13    Public Sub New()
14        MyBase.New("Get Element Groups", "ElGroups", _
15            "Import Element Groups from GSA File", _
16            "GsaComTools", "Elements")
17    End Sub
18
19    ''' <summary>
20    ''' Registers all the input parameters for this component.
21    ''' </summary>
22    Protected Overrides Sub RegisterInputParams(pManager As GH_Component.GH_InputParamManager)
23        pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
24        pManager.AddIntegerParameter("Elements", "Els", "Element Numbers", GH_ParamAccess.list)
25    End Sub
26
27    ''' <summary>
28    ''' Registers all the output parameters for this component.
29    ''' </summary>
30    Protected Overrides Sub RegisterOutputParams(pManager As GH_Component.GH_OutputParamManager)
31        pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
32        pManager.AddIntegerParameter("Element Groups", "ElGroups", "Element Groups", GH_ParamAccess.list)
33    End Sub
34
35    ''' <summary>
36    ''' This is the method that actually does the work.
37    ''' </summary>
38    ''' <param name="DA">The DA object is used to retrieve from inputs and store in outputs.</param>
39    Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)
40        'Declare Variables
41        Dim FileName As String = ""
42        Dim ElemRefs As New List(Of Integer)
43        Dim Elements() As GsaElement
44        Dim ElementGroups As New List(Of Integer)
45        Dim s As Short
46        Dim myDataCollector As New DataCollector
47        Dim myObjectWrapper As Types.GH_ObjectWrapper
48
49        'Import GSA File Name

```

```
50     If (Not DA.GetData(0, myObjectWrapper)) Then Return
51     If (Not DA.GetDataList(1, ElemRefs)) Then Return
52
53     Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)
54
55     'Get Data
56     s = GsaObj.Elements(ElemRefs.ToArray, Elements)
57     For Each item In Elements
58         ElementGroups.Add(item.Group)
59     Next
60
61     ' Assign the data to the output parameter.
62     DA.SetData(0, GsaObj)
63     DA.SetDataList(1, ElementGroups)
64
65     'Send Usage Data
66     myDataCollector.UsageStatistics("GetElementGroups")
67
68 End Sub
69
70 ''' <summary>
71 ''' Provides an Icon for every component that will be visible in the User Interface.
72 ''' Icons need to be 24x24 pixels.
73 ''' </summary>
74 Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
75     Get
76         'You can add image files to your project resources and access them like this:
77         ' return Resources.IconForThisComponent;
78         Return My.Resources.Element_ElGroups
79     End Get
80 End Property
81
82 ''' <summary>
83 ''' Gets the unique ID for this component. Do not change this ID after release.
84 ''' </summary>
85 Public Overrides ReadOnly Property ComponentGuid() As Guid
86     Get
87         Return New Guid("{3e18f5af-91e5-4bf3-8fc0-74af772eb681}")
88     End Get
89 End Property
90 End Class
```