

```

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 GetNodeNumbers
9     Inherits GH_Component
10     ''' <summary>
11     ''' Initializes a new instance of the GetNodeNumbers class.
12     ''' </summary>
13     Public Sub New()
14         MyBase.New("Get Node Numbers", "NodeNums", _
15             "Import Node Numbers from GSA File", _
16             "GsaComTools", "Nodes")
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.AddTextParameter("List", "List", "List name or number (all records will be returned if left blank)", GH_ParamAccess.item)
25
26         pManager(1).Optional = True
27     End Sub
28
29     ''' <summary>
30     ''' Registers all the output parameters for this component.
31     ''' </summary>
32     Protected Overrides Sub RegisterOutputParams(pManager As GH_Component.GH_OutputParamManager)
33         pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
34         pManager.AddIntegerParameter("Node Numbers", "NodeNums", "Node Numbers", GH_ParamAccess.list)
35     End Sub
36
37     ''' <summary>
38     ''' This is the method that actually does the work.
39     ''' </summary>
40     ''' <param name="DA">The DA object is used to retrieve from inputs and store in outputs.</param>
41     Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)
42         'Declare Variables
43         Dim FileName As String = ""
44         Dim NodeRefs() As Integer
45         Dim s As Short
46         Dim sList As String
47         Dim myDataCollector As New DataCollector
48         Dim myObjectWrapper As Types.GH_ObjectWrapper
49

```

```
50     'Import GSA File Name
51     If (Not DA.GetData(0, myObjectWrapper)) Then Return
52     If (Not DA.GetData(1, sList)) Then sList = "all"
53
54     Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)
55
56     'Get Entities
57     s = GsaObj.EntitiesInList(sList, GsaEntity.NODE, NodeRefs)
58
59     'Sort Array
60     Array.Sort(NodeRefs)
61
62     ' Assign the data to the output parameter.
63     DA.SetData(0, GsaObj)
64     DA.SetDataList(1, NodeRefs)
65
66     'Send Usage Data
67     myDataCollector.UsageStatistics("GetNodeNumbers")
68
69 End Sub
70
71 ''' <summary>
72 ''' Provides an Icon for every component that will be visible in the User  ➤
73     Interface.
74 ''' Icons need to be 24x24 pixels.
75 ''' </summary>
76 Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
77     Get
78         'You can add image files to your project resources and access them  ➤
79         ' like this:
80         ' return Resources.IconForThisComponent;
81         Return My.Resources.Node_Nums
82     End Get
83 End Property
84
85 ''' <summary>
86 ''' Gets the unique ID for this component. Do not change this ID after  ➤
87     release.
88 ''' </summary>
89 Public Overrides ReadOnly Property ComponentGuid() As Guid
90     Get
91         Return New Guid("{a07e98f9-9cb7-4d33-8eaa-9672d7815ad1}")
92     End Get
93 End Property
94 End Class
```