

```
1 Imports System.Collections.Generic
2
3 Imports Grasshopper.Kernel
4 Imports Rhino.Geometry
5 Imports Interop.gsa_8_7
6
7
8 Public Class AnalyzeGSA
9     Inherits GH_Component
10     ''' <summary>
11     ''' Initializes a new instance of the Analyze class.
12     ''' </summary>
13     Public Sub New()
14         MyBase.New("Analyze", "Analyze", _
15             "Analyze and Save GSA File", _
16             "GsaComTools", "Analysis")
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("Analysis Task", "Task", "Analysis Task (Optional)", GH_ParamAccess.item)
25
26
27         pManager(1).Optional = True
28     End Sub
29
30     ''' <summary>
31     ''' Registers all the output parameters for this component.
32     ''' </summary>
33     Protected Overrides Sub RegisterOutputParams(pManager As GH_Component.GH_OutputParamManager)
34         pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
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         Dim FileName As String
43         Dim Task As Integer
44         Dim s As Short
45         Dim IntResult As Short
46         Dim Result As Boolean
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.GetData(1, Task)) Then Task = 0
52
53     Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)
54
55     'Analyze
56     s = GsaObj.Analyse(CObj(Task))
57     IntResult = s
58
59     Select Case s
60     Case 1
61         AddRuntimeMessage(GH_RuntimeMessageLevel.Warning, "No GSA file
62             is open")
63     Case 2
64         AddRuntimeMessage(GH_RuntimeMessageLevel.Warning, "Failed to
65             attempt analysis")
66     End Select
67
68     'Return Results
69     DA.SetData(0, GsaObj)
70
71 End Sub
72
73 ''' <summary>
74 ''' Provides an Icon for every component that will be visible in the User
75 Interface.
76 ''' Icons need to be 24x24 pixels.
77 ''' </summary>
78 Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
79     Get
80         'You can add image files to your project resources and access them
81         like this:
82         ' return Resources.IconForThisComponent;
83         Return My.Resources.Analysis_Analyze
84     End Get
85 End Property
86
87 ''' <summary>
88 ''' Gets the unique ID for this component. Do not change this ID after
89 release.
90 ''' </summary>
91 Public Overrides ReadOnly Property ComponentGuid() As Guid
92     Get
93         Return New Guid("{3657ec0e-4836-41d5-bdaa-bc68762f3862}")
94     End Get
95 End Property
96 End Class
```