```vb
 1  Imports System.Collections.Generic
 2
 3  Imports Grasshopper.Kernel
 4  Imports Grasshopper.Kernel.Data
 5  Imports Grasshopper.Kernel.Types
 6  Imports Rhino.Geometry
 7  Imports Interop.gsa_8_7
 8  Imports UsageDataCollection
 9  Imports System.Runtime.InteropServices
10
11
12  Public Class CutAssemblyOutputExtract
13      Inherits GH_Component
14      ''' <summary>
15      ''' Initializes a new instance of the CutAssemblyOutputExtract class.
16      ''' </summary>
17      Public Sub New()
18          MyBase.New("Output Extract Cut Assembly", "CutAssembly", _
19                      "Fetches 'Cut Section Forces' results for a given
                        assembly", _
20                      "GsaComTools", "Misc")
21      End Sub
22
23      Enum Output_Init_Flags
24          OP_INIT_2D_BOTTOM = &H1       ' output 2D stresses at bottom layer
25          OP_INIT_2D_MIDDLE = &H2       ' output 2D stresses at middle layer
26          OP_INIT_2D_TOP = &H4          ' output 2D stresses at top layer
27          OP_INIT_2D_BENDING = &H8      ' output 2D stresses at bending layer
28          OP_INIT_2D_AVGE = &H10        ' average 2D element stresses at nodes
29          OP_INIT_1D_AUTO_PTS = &H20    ' calculate 1D results at interesting
                points
30          OP_INIT_INFINITY = &H40       ' return infinity and NaN values as such,
                else as zero
31      End Enum
32      Enum Output_IsDataRef_Flags
33          OP_IS_AND = &H1               ' otherwise OR
34          OP_IS_PER_REC = &H2
35          OP_IS_PER_NODE = &H4
36          OP_IS_PER_ELEM = &H8
37          OP_IS_PER_MEMB = &H10
38          OP_IS_PER_1D_DISP = &H20
39          OP_IS_PER_1D_FRC = &H40
40          OP_IS_PER_TOPO = &H80
41          OP_IS_AT_CENTRE = &H100
42      End Enum
43
44      ''' <summary>
45      ''' Registers all the input parameters for this component.
46      ''' </summary>
47      Protected Overrides Sub RegisterInputParams(pManager As
          GH_Component.GH_InputParamManager)
48          pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object",
              GH_ParamAccess.item)
49          pManager.AddIntegerParameter("Assembly Ref", "Assembly", "The assembly
                to extract forces for", GH_ParamAccess.item)
50          pManager.AddBooleanParameter("Average 2D Stresses", "Avg2DStress",
```

```vb
                       "Use Averaged 2D Stresses", GH_ParamAccess.item)
51          pManager.AddNumberParameter("Start", "Start", "First offset (set to 0 ⮑
                if no offsets are to be defined, optional)", GH_ParamAccess.item)
52          pManager.AddNumberParameter("End", "End", "Last offset (set to 0 if no ⮑
                 offsets are to be defined, optional)", GH_ParamAccess.item)
53          pManager.AddTextParameter("Case", "Case", "Case description",            ⮑
                GH_ParamAccess.item)
54          pManager.AddTextParameter("Axis", "Axis", "Axis definition              ⮑
                (optional)", GH_ParamAccess.item)
55
56          pManager(3).Optional = True
57          pManager(4).Optional = True
58          pManager(6).Optional = True
59      End Sub
60
61      ''' <summary>
62      ''' Registers all the output parameters for this component.
63      ''' </summary>
64      Protected Overrides Sub RegisterOutputParams(pManager As              ⮑
          GH_Component.GH_OutputParamManager)
65          pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object",            ⮑
                GH_ParamAccess.item)
66          pManager.AddNumberParameter("Cut Section Forces", "Forces", "Cut        ⮑
                Section Forces (Branches are x, y, z, xx, yy, zz)",               ⮑
                GH_ParamAccess.tree)
67          pManager.AddNumberParameter("Cut Section Positions", "Positions", "Cut ⮑
                 Section Positions", GH_ParamAccess.list)
68      End Sub
69
70      ''' <summary>
71      ''' This is the method that actually does the work.
72      ''' </summary>
73      ''' <param name="DA">The DA object is used to retrieve from inputs and      ⮑
          store in outputs.</param>
74
75      Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)
76          'Declare Variables
77          Dim myOutput() As Interop.gsa_8_7.GsaResults = Nothing
78          Dim myPositions As New List(Of Double)
79          Dim Output As New Grasshopper.DataTree(Of Double)
80          Dim s As Short
81          Dim Axis As String = Nothing
82          Dim LoadCase As String = Nothing
83          Dim i, j, k As Integer
84          Dim myDataCollector As New DataCollector
85          Dim myObjectWrapper As Types.GH_ObjectWrapper = Nothing
86          Dim AssemblyRef As Integer
87          Dim Avg2DStress As Boolean
88          Dim OffStart, OffEnd As Double
89
90          'Import GSA File Name
91          If (Not DA.GetData(0, myObjectWrapper)) Then Return
92          If (Not DA.GetData(1, AssemblyRef)) Then Return
93          If (Not DA.GetData(2, Avg2DStress)) Then Return
94          If (Not DA.GetData(3, OffStart)) Then OffStart = 0
95          If (Not DA.GetData(4, OffEnd)) Then OffEnd = 0
```

```vb
 96            If (Not DA.GetData(5, LoadCase)) Then Return
 97            If (Not DA.GetData(6, Axis)) Then Axis = "Global"
 98
 99            Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)
100
101            'Get Data
102            s = GsaObj.Output_Extract_CutAssembly(AssemblyRef, Avg2DStress,
                 LoadCase, Axis, myOutput)
103
104            For j = 0 To 5
105                Output.AddRange(New List(Of Double), New GH_Path(j))
106                For i = 0 To myOutput.Length - 1
107                    Output.Branches.Item(j).Add(CDbl(myOutput(i).dynaResults
                       (j).ToString))
108                Next
109            Next
110
111            For i = 0 To myOutput.Length - 1
112                myPositions.Add(CDbl(myOutput(i).Pos))
113            Next
114
115            ' Assign the data to the output parameter.
116            DA.SetData(0, GsaObj)
117            DA.SetDataTree(1, Output)
118            DA.SetDataList(2, myPositions)
119
120            'Send Usage Data
121            myDataCollector.UsageStatistics("CutAssemblyOutputExtract")
122
123        End Sub
124
125        ''' <summary>
126        ''' Provides an Icon for every component that will be visible in the User
             Interface.
127        ''' Icons need to be 24x24 pixels.
128        ''' </summary>
129        Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
130            Get
131                'You can add image files to your project resources and access them
                   like this:
132                ' return Resources.IconForThisComponent;
133                Return My.Resources.Assembly_CutSecForces
134            End Get
135        End Property
136
137        ''' <summary>
138        ''' Gets the unique ID for this component. Do not change this ID after
             release.
139        ''' </summary>
140        Public Overrides ReadOnly Property ComponentGuid() As Guid
141            Get
142                Return New Guid("{43ac9153-eb10-458d-b2f3-f129165ec334}")
143            End Get
144        End Property
145 End Class
```