

```

1 Imports System.Collections.Generic
2 Imports System.Linq
3 Imports Grasshopper.Kernel
4 Imports Rhino.Geometry
5 Imports UsageDataCollection
6
7 Public Class Record
8     Public Property index As Int32
9     Public Property value As String
10 End Class
11
12 Public Class GetIndexOfRecord
13     Inherits GH_Component
14     ''' <summary>
15     ''' Initializes a new instance of the GetIndexOfRecord class.
16     ''' </summary>
17     Public Sub New()
18         MyBase.New("Get Index of Record", "RecIndex", _
19             "Find Index of Specific Record in List. Optionally Select", _
20             "Corresponding Data from Other List", _
21             "GsaComTools", "Data")
22     End Sub
23     ''' <summary>
24     ''' Registers all the input parameters for this component.
25     ''' </summary>
26     Protected Overrides Sub RegisterInputParams(pManager As GH_Component.GH_InputParamManager)
27         pManager.AddTextParameter("Record Set", "Records", "List of records to search in.", GH_ParamAccess.list)
28         pManager.AddTextParameter("Record", "Record", "Record to search for.", GH_ParamAccess.list)
29         pManager.AddGenericParameter("{Data}", "{Data}", "Optional corresponding data list.", GH_ParamAccess.list)
30
31         pManager(2).Optional = True
32     End Sub
33
34     ''' <summary>
35     ''' Registers all the output parameters for this component.
36     ''' </summary>
37     Protected Overrides Sub RegisterOutputParams(pManager As GH_Component.GH_OutputParamManager)
38         pManager.AddBooleanParameter("Success", "Success", "Success", GH_ParamAccess.item)
39         pManager.AddIntegerParameter("Index", "Index", "Index of record within 'Records' set", GH_ParamAccess.list)
40         pManager.AddGenericParameter("Data", "Data", "Corresponding data from data set, if any", GH_ParamAccess.list)
41     End Sub
42
43     ''' <summary>
44     ''' This is the method that actually does the work.
45     ''' </summary>
46     ''' <param name="DA">The DA object is used to retrieve from inputs and store in outputs.</param>

```

```
47     Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)
48
49         Dim RecordSet As New List(Of String)
50         Dim Records As New List(Of String)
51         Dim Data As New List(Of Object)
52         Dim Indices As New List(Of Integer)
53         Dim Output As New List(Of Object)
54         Dim myRecord As Record
55         Dim i As Int32
56         Dim NoResults As Boolean
57         Dim myIndex As Int32
58         Dim myDataCollector As New DataCollector
59
60         'Get Inputs
61         If (Not DA.GetDataList(0, RecordSet)) Then Return
62         If (Not DA.GetDataList(1, Records)) Then Return
63         If (Not DA.GetDataList(2, Data)) Then Data = Nothing
64
65         'For Each Record In Records
66         '     Indices.Add(RecordSet.IndexOf(Record))
67         'Next
68
69         Dim arrList As New ArrayList()
70
71         i = 0
72         For Each item In RecordSet
73             myRecord = New Record With {.index = i, .value = item}
74             arrList.Add(myRecord)
75             i = i + 1
76         Next
77
78         Dim query
79
80         For Each item In Records
81             NoResults = True
82             query = From o As Record In arrList
83                     Where o.value = item
84                     Select o
85                     Take 1
86             For Each p As Record In query
87                 Indices.Add(p.index)
88                 NoResults = False
89             Next
90             If NoResults = True Then Indices.Add(-1)
91         Next
92
93
94         'Get indices for each record
95         'For Each item In Records
96         '    i = 0
97         '    Do
98         '        If i = RecordSet.Count Then
99         '            Indices.Add(Nothing)
100        '        Exit Do
101        '        ElseIf item = RecordSet.Item(i) Then
102        '            Indices.Add(i)
```

```
103         '           Exit Do
104         '           End If
105         '           i = i + 1
106         '       Loop
107     'Next
108
109     'If data exists, get associated data
110     If Data IsNot Nothing Then
111         For Each item In Indices
112             Output.Add(Data.Item(item))
113         Next
114     Else
115         Output = Nothing
116     End If
117
118     'Send Usage Data
119     myDataCollector.UsageStatistics("GetIndexOfRecord")
120
121     'Set output variables
122     DA.SetData(0, True)
123     DA.SetDataList(1, Indices)
124     DA.SetDataList(2, Output)
125
126
127 End Sub
128
129 ''' <summary>
130 ''' Provides an Icon for every component that will be visible in the User ↗
131     Interface.
132 ''' Icons need to be 24x24 pixels.
133 ''' </summary>
134 Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
135     Get
136         'You can add image files to your project resources and access them ↗
137         ' like this:
138         ' return Resources.IconForThisComponent;
139         Return My.Resources.GetIndexOfRecord
140     End Get
141 End Property
142
143 ''' <summary>
144 ''' Gets the unique ID for this component. Do not change this ID after ↗
145     release.
146 ''' </summary>
147 Public Overrides ReadOnly Property ComponentGuid() As Guid
148     Get
149         Return New Guid("{28ab056c-c0de-4bdc-9a03-4f07cd7910e1}")
150     End Get
151 End Property
152 End Class
```