```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using Autodesk.Revit;
5  using Autodesk.Revit.Elements;
6  using System.Diagnostics;
7  using System.Windows.Forms;
8
9  namespace CarbonMapping
10 {
11     public class CarbonMapping:IExternalCommand
12     {
13         public IExternalCommand.Result Execute(ExternalCommandData commandData, ref string message, ElementSet elements)
14         {
15             Autodesk.Revit.Application revitApp = commandData.Application;
16             Document doc = revitApp.ActiveDocument;
17
18             Dictionary<string, double> walls = new Dictionary<string, double>();
19
20             ElementIterator iter = doc.Elements;
21             iter.Reset();
22
23             int count = 0;
24             string outString = "";
25
26             // iterate through each element in the Document
27             while (iter.MoveNext())
28             {
29                 Wall w = iter.Current as Wall;
30
31                 if (w != null)
32                 {
33                     double v = getParameterByName(w, "Volume");
34                     walls.Add(w.WallType.Name, v);
35
36                     outString += ("wall Type Name:[ " + w.WallType.Name + " ]  |  ");
37                     outString += ("wall Type UniqueId:[ " + w.WallType.UniqueId + " ]  |  ");
38                     outString += ("\nwallID:[ " + w.UniqueId + " ]  |  ");
39                     outString += ("wall Volume:[ " + v + " ]\n\n");
40                 }
41                 else
42                 {
43                     count++;
44                 }
45             }
46
47
48             //Print out for Debugging
49             Debug.Print("\nwall element Size : " + walls.Count);
50             Debug.Print(("\nElement Size : " + count));
51             Debug.Print(outString);
52
```

```
53              string msString = ("\nwall element Size : " + walls.Count) + ("\nElement Size : " + count +"\n") + outString;
54              MessageBox.Show(msString);
55
56              return IExternalCommand.Result.Succeeded;
57          }
58
59          /// <summary>
60          /// Function to retrieve Parameter value by their Name
61          /// </summary>
62          /// <param name="e"> Element </param>
63          /// <param name="name"> Parameter Definition Name</param>
64          /// <returns> double value will be returned.</returns>
65          public double getParameterByName(Element e, string name)
66          {
67              foreach(Parameter p in e.Parameters)
68              {
69                  if (p.Definition.Name == name)
70                      return p.AsDouble();
71              }
72              return 0.0;
73          }
74      }
75  }
76
```