

C:\Documents and Settings\Administrator\My Documents\Visual...Projects\Exercise10282009\Exercise10282009\wallLayerInfo.cs 1

```
using System;
using System.Collections.Generic;
using System.Text;

using Autodesk.Revit;
using Autodesk.Revit.Parameters;
using Autodesk.Revit.Elements;

using System.Windows.Forms;
using System.Diagnostics;

namespace Exercise10282009
{
    class wallLayer : IExternalCommand
    {
        public IExternalCommand.Result Execute(ExternalCommandData revit, ref String message, ElementSet elements)
        {
            // add references to the Revit Application and Document objects for later use
            Autodesk.Revit.Application m_app = revit.Application;
            Document r_doc = m_app.ActiveDocument;

            // initiate a null Wall object
            Wall wall = null;

            // Wall element counter
            int counter = 0;

            // iterate through the element selection in the Document
            foreach (Element e in r_doc.Selection.Elements)
            {
                wall = e as Wall; // cast a element object as a Wall object

                // initiate the information string
                String what = "Compound in Layers: \n";
                if (wall != null) // check if wall is not nulls
                {
                    counter++; //increase the number of counter

                    // add the information of layer size to the information string
                    what += ("CompoundLayers size: " + wall.WallType.CompoundStructure.Layers.Size + "\n");

                    what += ("Wall width: " + wall.Width + "\n"); // add wall width

                    // retrieve the parameter-Area by using get_Parameter() function
                    Parameter p_area = wall.get_Parameter(BuiltInParameter.HOST_AREA_COMPUTED);
                    what += ("Area: " + p_area.AsDouble() + "\n");
                }
            }
        }
    }
}
```

C:\Documents and Settings\Administrator\My Documents\Visual...Projects\Exercise10282009\Exercise10282009\wallLayerInfo.cs 2

```
// retrieve the parameter-volume by using get_Parameter() function
Parameter p_volume = wall.get_Parameter(BuiltInParameter.HOST_VOLUME_COMPUTED);
what += ("Volume: " + p_volume.AsDouble() + "\n");

// accumulate the volume info for comparison.
// initial accumulated volume = 0.0
double totalVolumn = 0.0;

// check each layer of the compoundStructure
for (int j = 0; j < wall.WallType.CompoundStructure.Layers.Size; j++)
{
    try
    {
        // get layer material name
        what += ("material Name : " + wall.WallType.CompoundStructure.Layers.get_Item(j).Material.
Name);
    }
    catch { what += "no material "; }

    what += " | Thickness: ";
    // get layer thickness
    double t = wall.WallType.CompoundStructure.Layers.get_Item(j).Thickness;
    what += wall.WallType.CompoundStructure.Layers.get_Item(j).Thickness;
    what += "\n";

    // accumulate the volume by layer
    totalVolumn += t * p_area.AsDouble();
}

what += ("accumulated volume from layers: " + totalVolumn);

}
MessageBox.Show(what);
//Debug.Print(what);
}

return IExternalCommand.Result.Succeeded;
}
}
```