

```

1  using System;
2  using System.Windows.Forms;
3  using SldWorks;
4  using System.Runtime.InteropServices;
5  using Excel = Microsoft.Office.Interop.Excel;
6
7  namespace SW_Macro_Excel
8  {
9      public partial class Form1 : Form
10     {
11         SldWorks.SldWorks swApp;
12         Excel.Application exlApp;
13         Excel.Workbook exlBook;
14         Excel.Worksheet exlSheet;
15
16         int index = 2;
17
18         public Form1()
19         {
20             InitializeComponent();
21
22             try {
23                 swApp = (SldWorks.SldWorks)Marshal.GetActiveObject
24                     ("SldWorks.Application");
25             }
26             catch (Exception) {
27                 swApp = new SldWorks.SldWorks();
28                 swApp.Visible = true;
29             }
30
31             private void b_analyze_Click(object sender, EventArgs e)
32             {
33                 // Entleeren der ListBox
34                 lb_output.Items.Clear();
35
36                 ModelDoc2 swModel = swApp.ActiveDoc;
37                 Configuration ACTC = swModel.GetActiveConfiguration();
38                 Component2 root = ACTC.GetRootComponent();
39
40                 // Verbindung zu Excel herstellen
41                 exlApp = new Microsoft.Office.Interop.Excel.Application();
42                 exlApp.Visible = true;
43
44                 // neues Worksheet anlegen
45                 exlBook = exlApp.Workbooks.Add();
46                 exlSheet = exlBook.ActiveSheet;
47
48                 // Tabellenkopf anlegen
49                 exlSheet.Cells[1, 1] = "Menge";
50                 exlSheet.Cells[1, 2] = "Einheit";
51                 exlSheet.Cells[1, 3] = "Name";
52                 exlSheet.Cells[1, 4] = "Bemerkung";

```

```

53
54     // calling traverse Methode
55     traverse(root);
56 }
57
58 private void traverse(Component2 comp)
59 {
60     object[] childrens = comp.GetChildren();
61     foreach (Component2 child in childrens) {
62         traverse(child);
63     }
64
65     ModelDoc2 mdocChild = comp.GetModelDoc2();
66
67     if (mdocChild.GetType() == 1) {
68         // Bezeichnung kann über 2 Möglichkeiten erfolgen
69         // Dateiname (mit GetFileName wird nur der Dateiname aus      ➤
70             einem Dateipfad extrahiert
71         string entry = System.IO.Path.GetFileName(comp.GetPathName  ➤
72             ());
73         // oder
74         string entry2 = comp.Name2;
75         //entfernen der 2 hinteren Zeichen im String
76         entry2 = entry2.Remove(entry2.Length - 2);
77
78         int Row = 0;
79
80         // überprüfen, ob aktueller Eintrag bereits vorhanden ist, ➤
81         // wenn ja Zeilen-Index speichern
82         for (int i = 2; i < index; i++) {
83             if ((string)exlSheet.Cells[i, 3].Value2 == entry) {
84                 Row = i;
85             }
86         }
87         // wenn Zeilen Index gesetzt wurde, Wert für Anzahl aus ➤
88         // Zeile lesen und um 1 erhöhen
89         // erhöhten Wert in Zelle schreiben
90         if (Row > 0) {
91             double newCount = (double)exlSheet.Cells[Row, 1].Value2 ➤
92                 + 1;
93             exlSheet.Cells[Row, 1] = newCount;
94         }
95         else {
96             // Wenn ZeilenIndex nicht gesetzt wurde, neuen Dateipfad ➤
97
98             exlSheet.Cells[index, 1] = 1;
99             exlSheet.Cells[index, 2] = "Stück";
100             exlSheet.Cells[index, 3] = entry;
101
102             lb_output.Items.Add(entry);
103
104             index++;
105         }
106     }
107 }

```

```
100      }
101    }
102  }
103 }
104
```