

Hochschule Harz	FB Automatisierung und Informatik
Versuch: Excel-Export	Grafische Benutzer Oberflächen mit .net Thema: Dialogfenster mit dem Export nach Excel

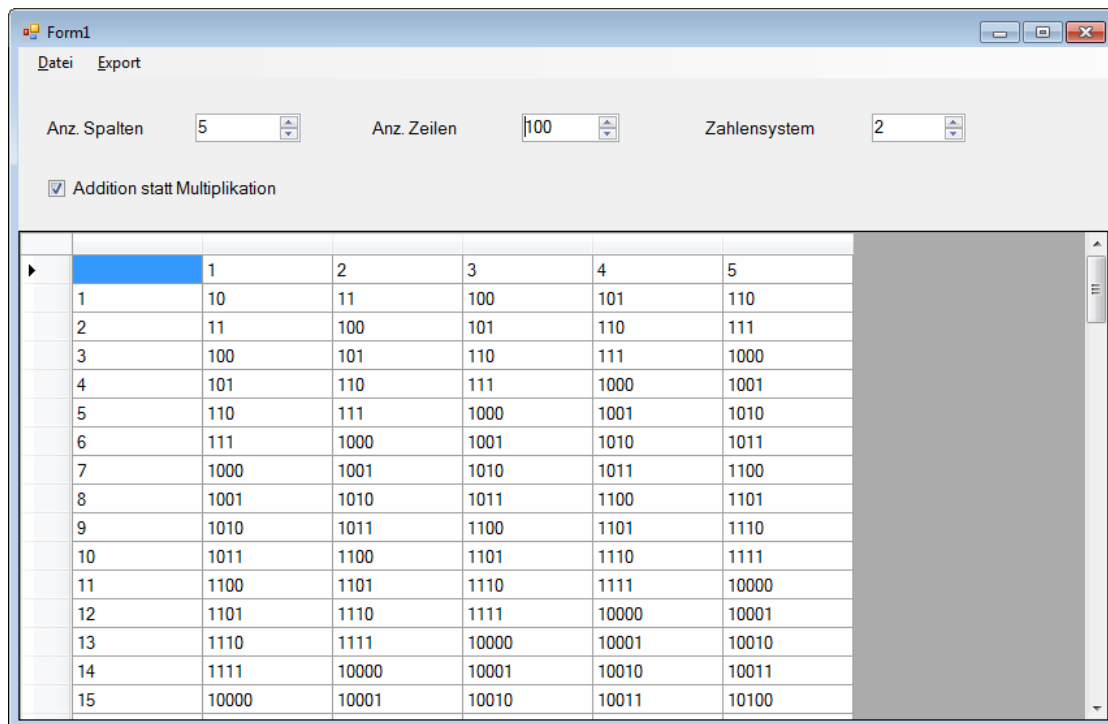
Versuchsziele

Vertiefung in der .net Programmierung mit dem Export nach Excel mittels COM-Technologien und der BIFF-Schnittstelle

1. Aufgabenstellung:

Entwickeln Sie ein SDI-Programm mit folgenden Eigenschaften:

- Darstellung einer Tabelle mit den spalten- und zeilenweise Additionen bzw. Multiplikationen in verschiedenen Zahlensystemen
- Das Fenster hat sieben GUI-Elemente (drei Labels, drei NumerikSpinUpDown und eine Checkbox)
- Dazu noch zwei Hauptmenüs



Ablauf

Projekt erstellen:

- Erstellen eines neuen Projektes (C#, Windows, Windows Forms Application). Der Name muss „Labor4“ sein
- Eintragen eines Panels (DockStyle = Top)
- Eintragen der sieben GUI-Elemente in das Panel

GUI-Element im Panel	Beschriftung / Name	Min/Max
Label	Anz. Spalten	5, 100
NumerikUpDown	nupColumns	
Label	Anz. Zeilen	5, 100
NumerikUpDown	nupRows	
Label	Zahlensystem	

NumerikUpDown	nupBasis	2,32
CheckBox	chAddition	

- Einfügen eines weiteren Panels, Name panelMitte
- Einfügen eines DataGridViews, Name: „Grid1“ in das Panel panelMitte
- In Form_Load DockStyle auf Fill setzen
panelMitte.Dock = DockStyle.Fill;
Grid1.Dock = DockStyle.Fill;
- **Weitere Information:**
- Grid1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
- Grid1.SelectionMode = DataGridViewSelectionMode.RowHeaderSelect;
- Grid1.ReadOnly = false;
- Grid1.AutoSize = true;
- Grid1.AllowUserToDeleteRows = false;

Aktionen in der form1 programmieren

- Erstellen Sie die ChangeEvent-Methoden für die drei NumerikUpDown Elemente bzw. für die CheckBox
- Erstellen Sie dafür eine neue Methode createTabelle, diese wird dann von allen drei Event aufgerufen
- Benutzen Sie für die Ausgabe folgende Methode der Klasse String:
string sOutput = String.Format("{0:X}", value);
oder
die Methode IntToStr
- Grid1.RowCount = 2; // setzen der Anzahl der Zeilen
- Grid1.ColumnCount = 2; // setzen der Anzahl der Spalten
- Grid1[c,r].Value = "Hallo"; // setzen einer bestimmten Zelle
- Setzen Sie in der ersten Spalte den Parameter (a, 1. Summand)
- Setzen Sie in der ersten Zeile den Parameter (b, 2. Summand)

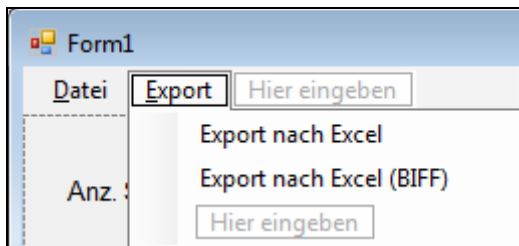
Hilfsfunktion:

```
private String IntToStr(int value, int basis) {
    String sStr;
    int rest, i;
    String[] abc = {"0","1","2","3","4","5","6","7","8","9",
                  "A","B","C","D","E","F","G","H","I","J",
                  "K","L","M","N","O","P","Q","R","S","T",
                  "U","V","W","X","Y","Z"};

    i=value;
    sStr="";
    if (i==0) {
        sStr="0";
    }
    else {
        while (i>0) {
            rest = i % basis;
            sStr=abc[rest]+sStr;
            i = i / basis;
        }
    }
    if (sStr.Length==1) sStr='0'+sStr;
    return sStr;
}
```

Export-Funktionen

- Einfügen eines Hauptmenüs im MainFrame, Name Export
- Einfügen folgender Einträge
 - Export nach Excel
 - Export nach Excel, BIFF



- Importieren der COM-Schnittstellen Excel
- Einfügen des BIFF-Exportcodes (siehe Homepage)
- Ändern bzw. überprüfen des Namespace !!

SaveDialog (nur Beispielcode)

```
SaveFileDialog saveFileDialog1 = new SaveFileDialog();
saveFileDialog1.Filter = "java files (*.java)|*.java|All files (*.*)|*.*";
saveFileDialog1.FilterIndex = 1;// zählt von 1 !
saveFileDialog1.DefaultExt = ".txt";
saveFileDialog1.InitialDirectory = "c:\\daten";
saveFileDialog1.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Personal);
saveFileDialog1.InitialDirectory = Environment.CurrentDirectory;
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    string sFilename = saveFileDialog1.FileName.ToString();
}
```

Export nach BIFF und Aufruf von Excel

```
using System.Diagnostics; // run
using Excel = Microsoft.Office.Interop.Excel;
```

Methode:

```
CExport2EXCEL exportexcel = new CExport2EXCEL("bsp3.xls");
exportexcel.openFile();
exportexcel.Write(1, 2, "abcde"); // col row string
// col row hier sollte ein String, KEIN Datum erscheinen, Problem CSV-Format
exportexcel.Write(1, 3, "01.01.2010");
// col row hier sollte ein String, KEIN Datum erscheinen, Problem CSV-Format
exportexcel.Write(1, 4, "01/01/2010");
exportexcel.closeFile();
Process Prog = new Process();
Prog.StartInfo.FileName = "excel.exe";
Prog.StartInfo.Arguments = "bsp3.xls";
Prog.Start();
```

Einfügen der COM-Verweise (Excel)

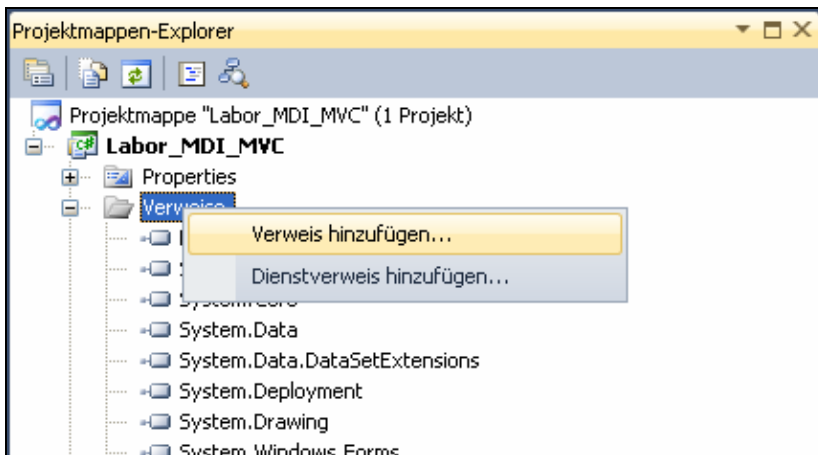
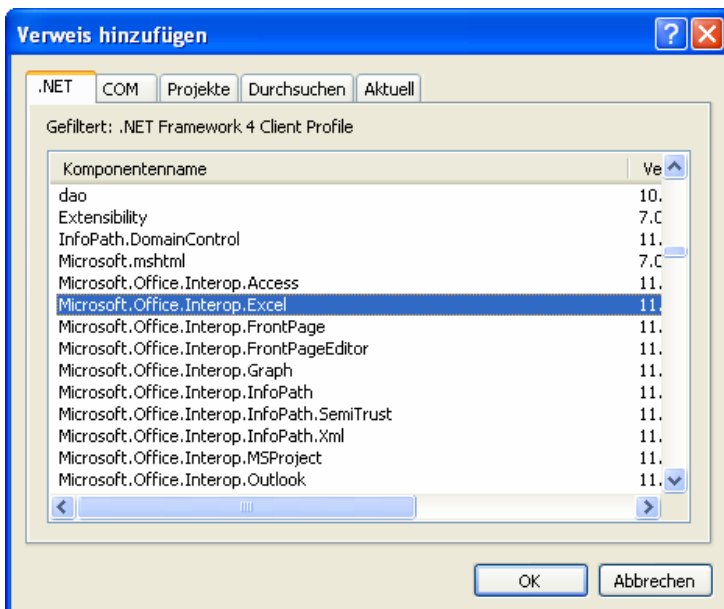


Abbildung 1 COM-Verweise ins Projekt hinzufügen

Auswahl der Verweise nach Excel: (ungefähr im obersten Drittel)



Beispielcode für Excel:

```
using Excel = Microsoft.Office.Interop.Excel;
```

```
var xl = new Excel.Application();
```

```
xl.Workbooks.Add();
```

```
xl.Visible = true;
```

```
xl.Cells[1, 1] = "Zahl1"; // Row, Column
```

```
xl.Cells[1, 2] = "Zahl2";
```

```
xl.Cells[2, 1] = "123";
```

```
xl.Cells[2, 2] = "345";
```

```
xl.Cells[4, 1].Select();
```

Range-Operation (Programmierersparnis):

```
xl.Range["A1:J1"].Font.Name = "Verdana";  
xl.Range["A1:J1"].Font.Color = Color.Red;  
xl.Range["A1:J1"].Font.Size = 14;  
xl.Range["A1:J1"].Font.Bold = true;  
xl.Range["A1:J1"].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;  
xl.Range["A1:J1"].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;  
xl.Range["A1:J1"].BorderAround(  
    LineStyle: Excel.XlLineStyle.xlDash,  
    Weight: Excel.XlBorderWeight.xlMedium  
);
```

Quellcode für die Events

- Einfügen des Event-Code in Form1 für Excel
Einbau des Quellcodes mittels COM
Bitte beachten Sie, dass Excel mit 1 startet !
Nach dem Öffnen muss der Cursor in Zelle B2 stehen

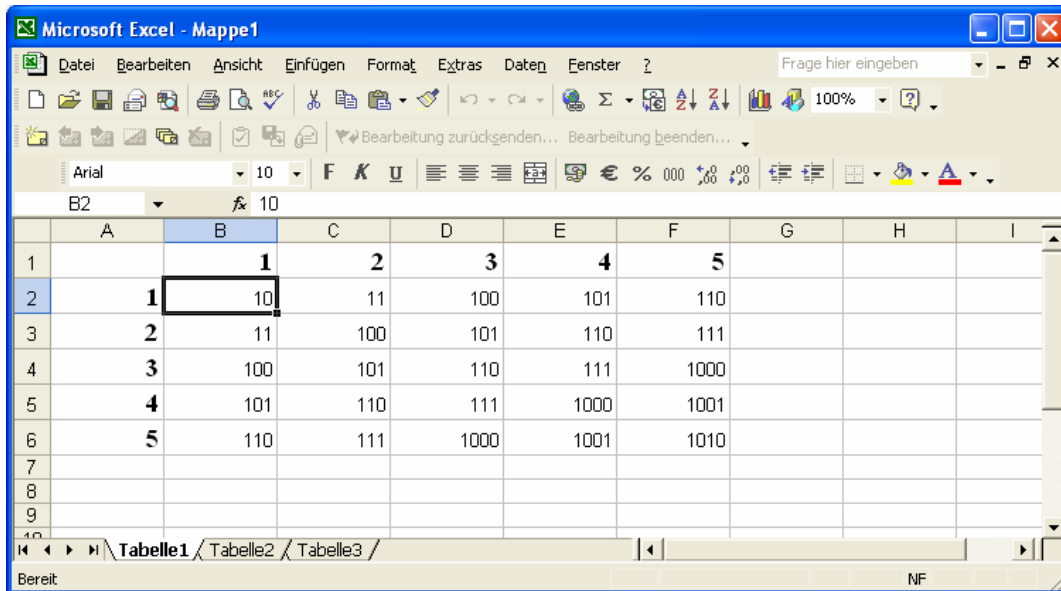


Abbildung 2 direkter Excelexport