

Grundlagen in Visual Studio MFC und .net

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- Raum 2.202
- Tel. 03943 / 659 338

Inhalt

1. Einführung in .net
- 2. Dialogfenster**
3. Grafik
4. SDI-Programme
5. Threads und Semaphore

Ein- und Ausgabe auf Konsolenebene

Klasse Console

■ Methoden:

Write("Text");

WriteLine("I: "+i.ToString() +Environment.NewLine + "Neue Zeile");

WriteLine("Index: "+i.ToString() +"\n" + "\tNeue Zeile");

Read(); // Charakter

Readln(); // string

string x = Console.ReadLine();

IsNumeric(sStr)

Convert.ToDecimal

ToInt32 ToInt64

Ein- und Ausgabe auf Konsolenebene

Formatausgabe mit WriteLine

```
Write("Text");
```

```
WriteLine("Index: "+i.ToString() +Environment.NewLine + "Neue Zeile" );
```

```
WriteLine("Index: "+i.ToString() +"\n" + "\tNeue Zeile" );
```

Formatausgaben:

C Währungsformat

D Dezimalformat

E Exponentialformat

F Festkommaformat

G Festkomma oder wissenschaftliche Notation

Ein- und Ausgabe auf Konsolenebene

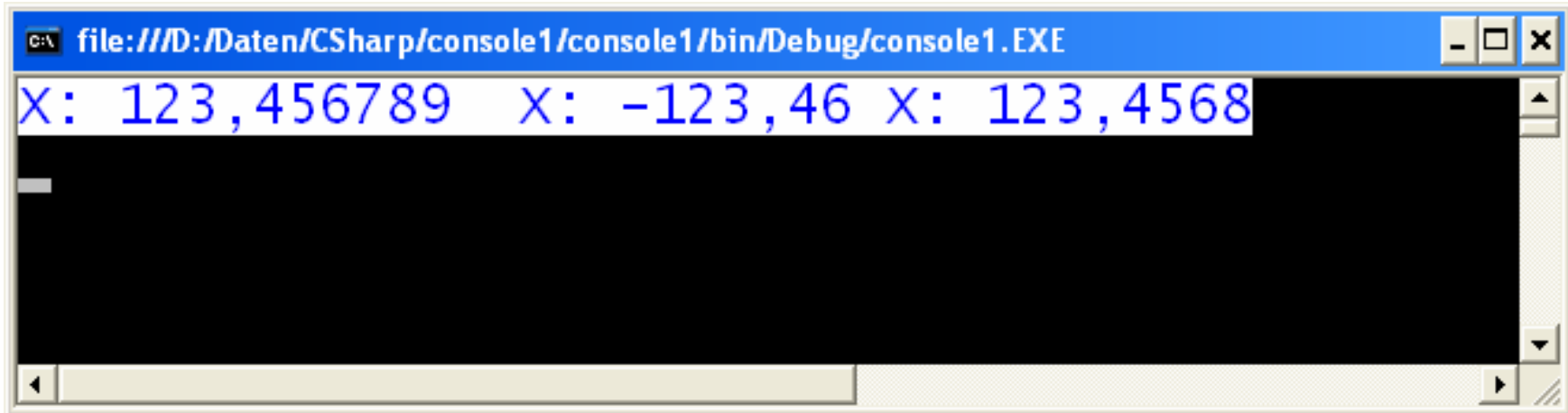
```
double x;
```

```
x = 123.456789;
```

```
double y;
```

```
y = -123.456789;
```

```
Console.WriteLine("X: {1} X: {0:#.###} X: {2:00#.0000}", y, x, x);
```



```
file:///D:/Daten/CSharp/console1/console1/bin/Debug/console1.EXE
X: 123,456789 X: -123,46 X: 123,4568
```

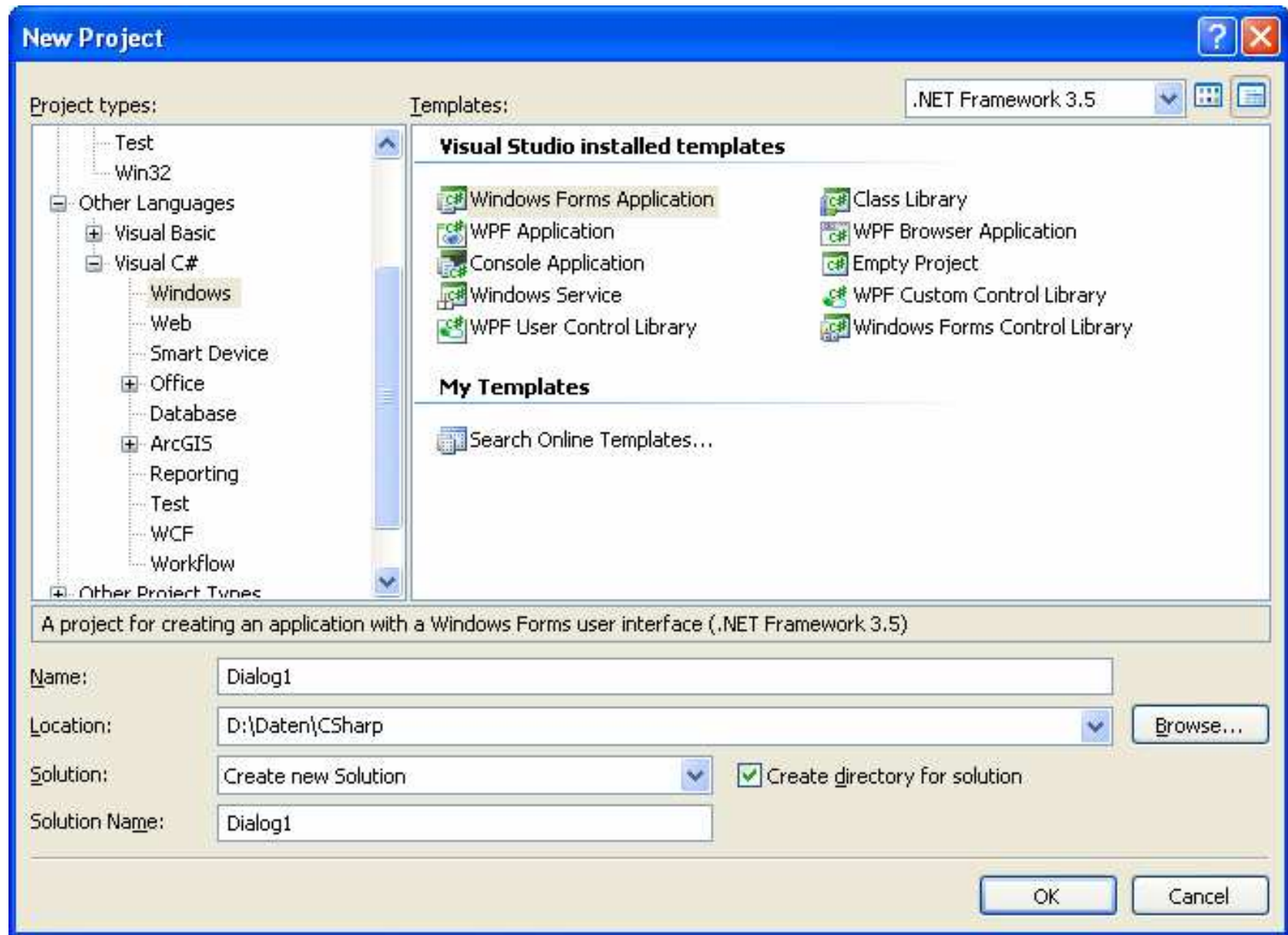
```

static void Main(string[] args) {
    string sStr;
    int a,b,c;
    Console.BackgroundColor = ConsoleColor.White;
    Console.ForegroundColor = ConsoleColor.Blue;
    Console.Write("1. Zahl eingeben: ");
    sStr = Console.ReadLine();
    if (IsNumeric(sStr))
    {
        a = Convert.ToInt32(sStr);
        Console.Write("2. Zahl eingeben: ");
        sStr = Console.ReadLine();
        if (IsNumeric(sStr))
        {
            b = Convert.ToInt32(sStr);
            c = a + b;
            Console.WriteLine("Summe: " + c.ToString());
            Console.Read();
        }
    }
}

```

IsNumeric

```
static bool IsNumeric(string sStr) {  
    int erg;  
    bool isNumber;  
    isNumber = Int32.TryParse(Convert.ToString(sStr),  
        System.Globalization.NumberStyles.Any,  
        System.Globalization.NumberFormatInfo.InvariantInfo,  
        out erg);  
    return isNumber;  
  
    }  
}
```



MessageBox

- Show(String)
- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator, Object)
- Show(Text, Caption)
- Show(IWin32Window, Text)
- Show(Text, Caption, MessageBoxButtons)
- Show(IWin32Window, Text, Caption)
- Show(Text, Caption, MessageBoxButtons, MessageBoxIcon)
- Show(IWin32Window, Text, Caption, MessageBoxButtons)
- Show(Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton)

MessageBox

- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon)
- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions)
- Show(Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, Text, Caption)
- Show(Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)

MessageBox

- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String)
- Show(Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator, Object)
- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, Text, Caption)
- Show(IWin32Window, Text, Caption, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)

Konstanten der MessageBox

MessageBoxButtons.YesNoCancel

MessageBoxButtons.YesNo

MessageBoxButtons.RetryCancel

MessageBoxButtons.OKCancel

MessageBoxButtons.AbortRetryIgnore

MessageBoxButtons.OK

DialogResult.Yes

DialogResult.OK

DialogResult.None

DialogResult.No

DialogResult.Ignore

DialogResult.Cancel

DialogResult.Abort

MessageBoxIcon.Warning

MessageBoxIcon.Stop

MessageBoxIcon.Question

MessageBoxIcon.None

MessageBoxIcon.Information

MessageBoxIcon.Hand

MessageBoxIcon.Exclamation

MessageBoxIcon.Error

MessageBoxIcon.Asterisk

■ Show(Text, Caption, MessageBoxButtons, MessageBoxIcon)

Konstanten der MessageBox

MessageBoxIcon.Warning

MessageBoxIcon.Stop

MessageBoxIcon.Question

MessageBoxIcon.None

MessageBoxIcon.Information

MessageBoxIcon.Hand

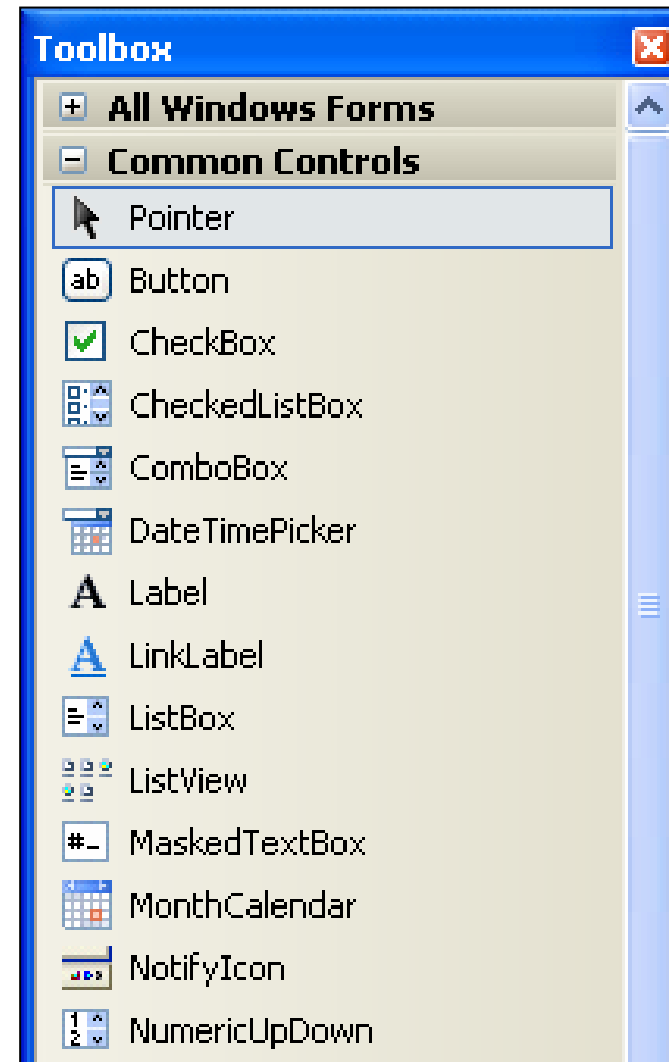
MessageBoxIcon.Exclamation

MessageBoxIcon.Error

MessageBoxIcon.Asterisk

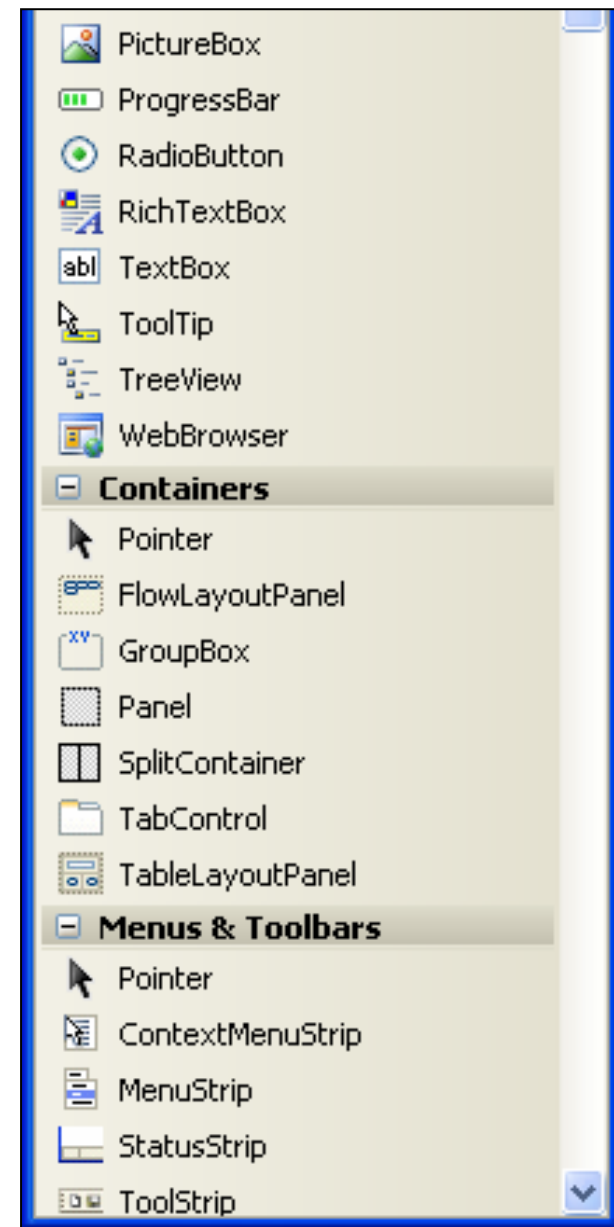
GUI-Elemente

- Button
- CheckBox
- CheckListBox
- DateTimePicker
- Label
- LinkLabel
- ListBox
- ListView
- MonthCalendar
- NotifyIcon
- NumericUpDown



GUI-Elemente

- PictureBox
- Progressbar
- RadioButton
- RichtextBox
- ToolTip
- TreeView
- WebBrowser
- GroupBox
- Panel
- SplitContainer
- TabControl
- TableLayoutPanel



Eigenschaften der GUI-Elemente: Label

- Name
- Anchor Top/Left
- BackColor
- BorderStyle
- Cursor
- Enabled False/True
- Font
- ForeColor
- Image
- ImageAlign
- Location: x y
- Locked True False
- Margin
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text: Beschriftung
- TextAlign: 9 Varianten
- Visible

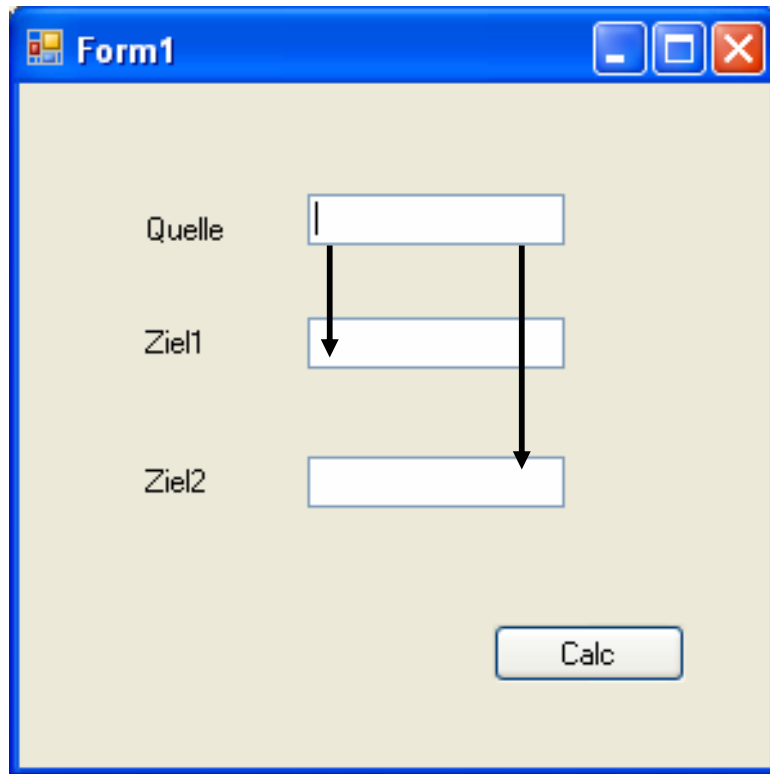
Eigenschaften der GUI-Elemente: Button

- Name
- Anchor Top/Left
- AutoSize: True/False
- BackColor
- Cursor
- DialogResult:
 - Ok Cancel
 - Abort Retry Ignore
 - Yes No
- Enabled False/True
- Font
- ForeColor
- Location: x y
- Locked True False
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text
- TextAlign: 9 Varianten
- Visible

Eigenschaften der GUI-Elemente: TextBox

- Name
- Anchor Top/Left
- BackColor
- Cursor
- Enabled False/True
- Font
- ForeColor
- Lines
- Multiline False/true
- Location: x y
- Locked True False
- PasswordChar
- ReadOnly False/True
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text: Inhalt
- Visible
- WordWrap False/True

1. Beispiel: Strings

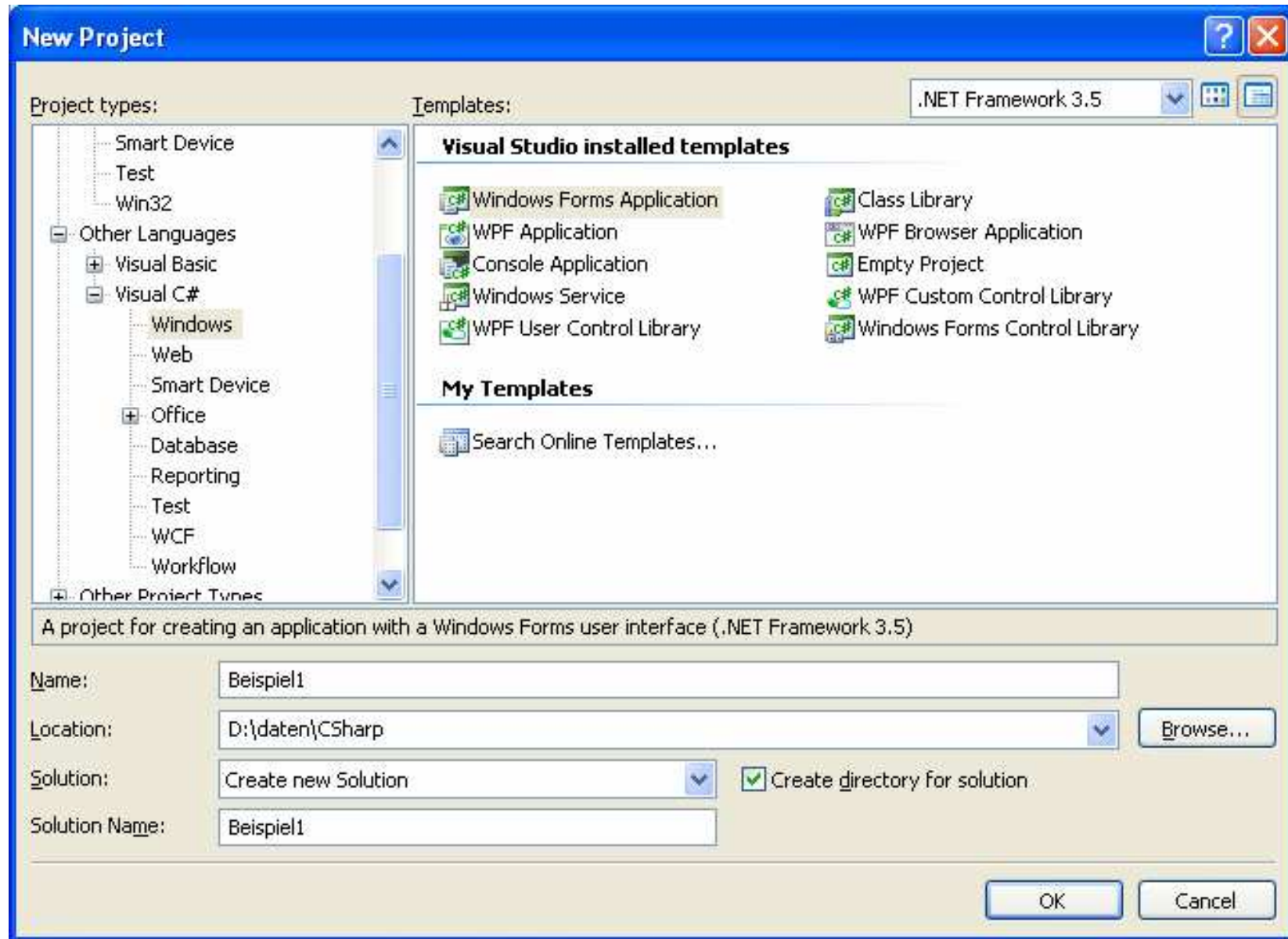


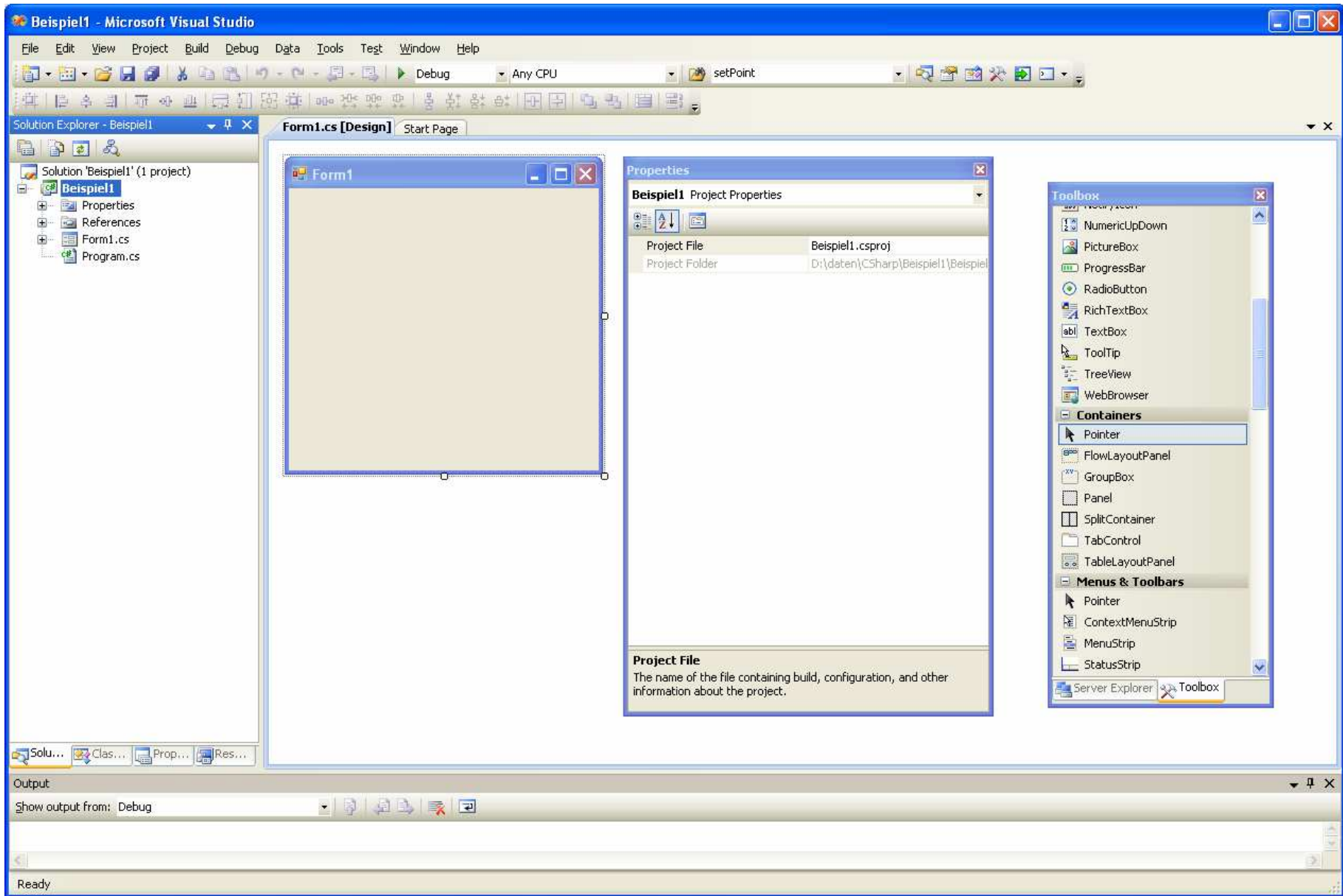
Namen:

- EQuelle
- EZiel1
- EZiel2
- BnCalc

Aufgaben

- Transportieren eines String per Schalter
- Transportieren eines String bei Änderung

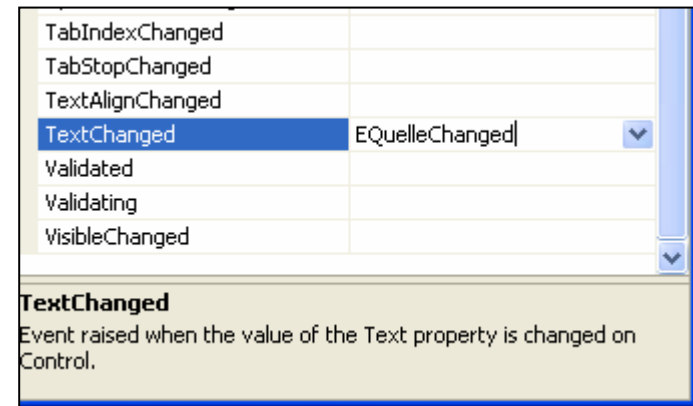




1. Beispiel

- Einbau der GUI-Elemente
 - Label Quelle
 - TextBox EQuelle
 - Label Ziel1
 - TextBox EZiel1
 - Label Ziel2
 - TextBox EZiel2
 - Schalter „Calc“ BnCalc
- Doppelklick auf den Schalter
- Eintragen des Quellcodes
- Einbauen der Event-Methode
- Eintragen des Quellcodes

```
EZiel1.Text = EQuelle.Text;
```

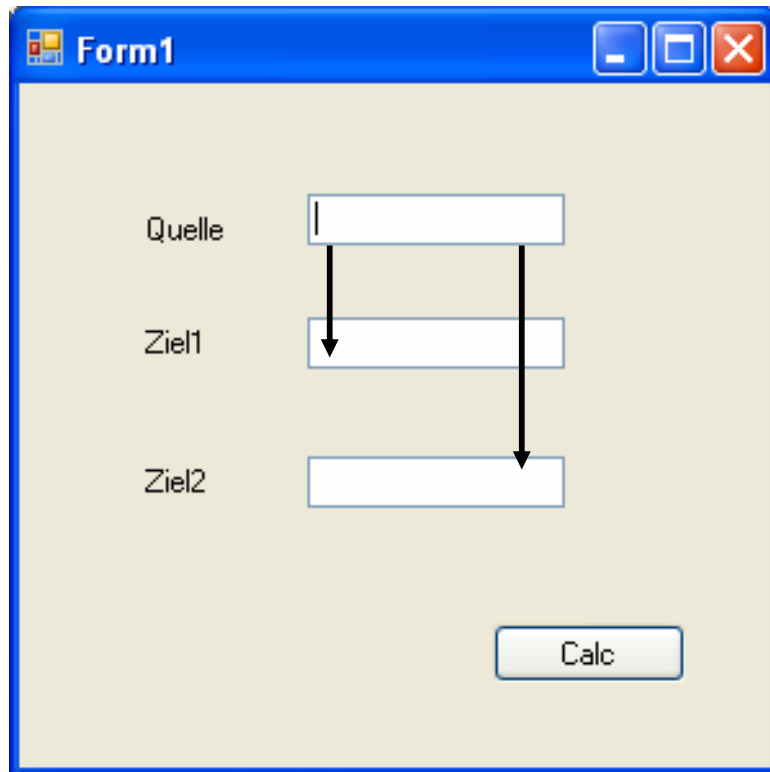


```
EZiel2.Text = EQuelle.Text;
```

Eigenschaften der GUI-Elemente: CheckBox

- Name
- Anchor Top/Left
- BackColor
- CheckAlign: 9 Varianten
- Checked
- Cursor
- Enabled False/True
- Font
- ForeColor
- Image
- ImageAlign
- Location: x y
- Locked True False
- Margin
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text: Beschriftung
- TextAlign: 9 Varianten
- ThreeState False/True, 3 Zustände
- Visible

2. Beispiel: Strings mit CheckBox



Namen:

- EQuelle
- EZiel1
- EZiel
- BnCalc
- ChkMove2Ziel2

Aufgaben

- Transportieren eines String per Schalter
- Transportieren eines String bei Änderung nur bei angeklickter CheckBox

2. Beispiel: Strings mit CheckBox

```
private void EQuelleChanged(object sender, EventArgs e)
{
    if ( ChkMove2Ziel2.Checked )
        EZiel2.Text = EQuelle.Text;
}
```

Eigenschaften der GUI-Elemente: GroupBox

- Name
- Anchor Top/Left
- BackColor
- BorderStyle
- Cursor
- Enabled False/True
- Font
- ForeColor
- Location: x y
- Locked True False
- Margin
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text: Beschriftung
- Visible

Eigenschaften der GUI-Elemente: RadioButton

- Name
- Anchor Top/Left
- AutoCheck
- BackColor
- CheckAlign 9 Varianten
- Checked
- Cursor
- DataBindings
- Enabled False/True
- Font
- ForeColor
- Image
- ImageAlign
- Location: x y
- Locked True False
- Margin
- Size: Width Height
- TabIndex
- TabStop
- Tag
- Text: Beschriftung
- TextAlign
- Visible

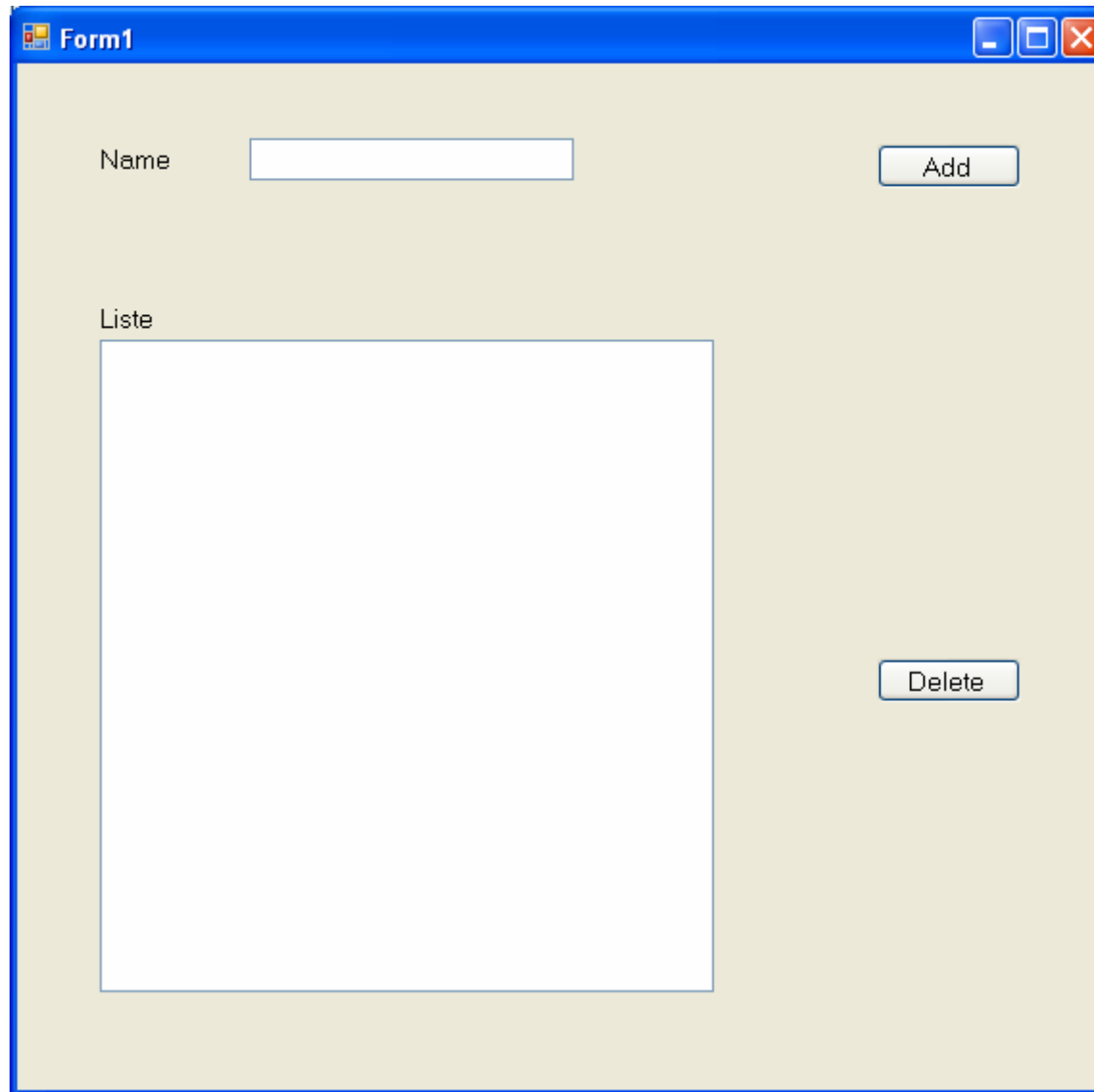
Eigenschaften der GUI-Elemente: ListBox

- Name
- Anchor Top/Left
- BackColor
- Cursor
- DataSource
- Enabled False/True
- Font
- ForeColor
- FormatString
- FormattingEnabled
- Items (Collection)
- Location: x y
- Locked True False
- Margin
- MultiColumn
- SelectionMode:
None, One,
MultiSimple, MultiExtended
- Size: Width Height
- TabIndex
- TabStop
- Tag
- UsesTabStop
- Text: Beschriftung
- TextAlign
- ValueMember (DBS)
- Visible

Eigenschaften der GUI-Elemente: ComboBox

- Name
- Anchor Top/Left
- BackColor
- Cursor
- **DropDownStyle:**
 - Simple, DropDown, DropDownList
- Enabled False/True
- Font
- ForeColor
- FormatString
- FormattingEnabled
- ItemHeight
- **Items (Collection)**
- Location: x y
- Locked True False
- Margin
- **MaxDropDownItems**
- Size: Width Height
- Sorted
- TabIndex
- TabStop
- Tag
- Text: Inhalt
- TextAlign
- Visible

3. Beispiel



- EName
 - BnAdd
 - BnDelete
 - Liste
-
- BnAdd_Click
Check Eintrag gültig
 - BnDelete_Click
Check Eintrag selektiert

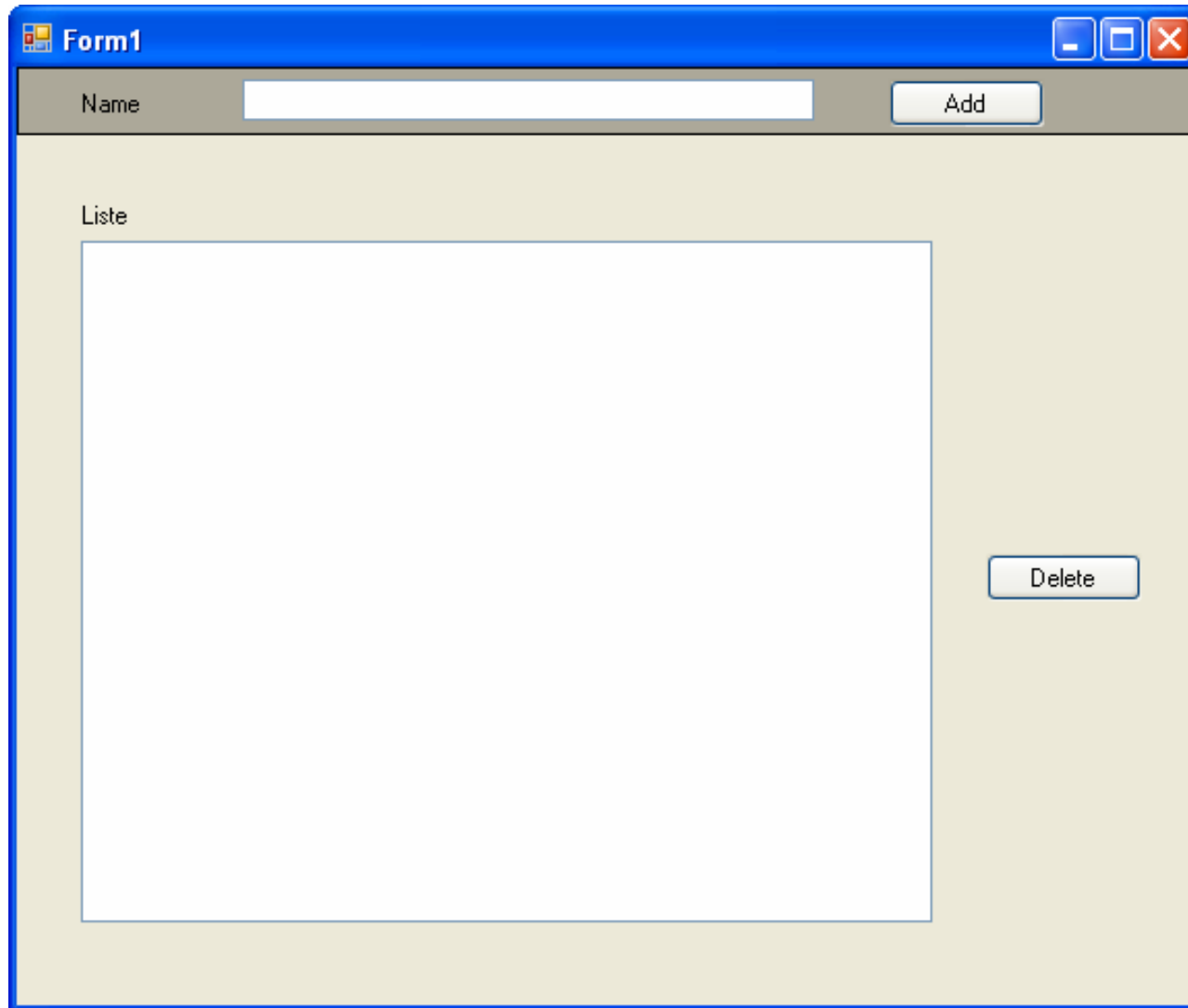
```
private void BnAdd_Click(object sender, EventArgs e)
{
    string sStr = EName.Text.Trim();
    if (sStr != "")
    {
        EName.Text = "";
        Liste.Items.Add(sStr);
    }
    else
    {
        MessageBox.Show("Bitte einen Eintrag in das Editorfeld");
        EName.Focus();
    }
}
```

```

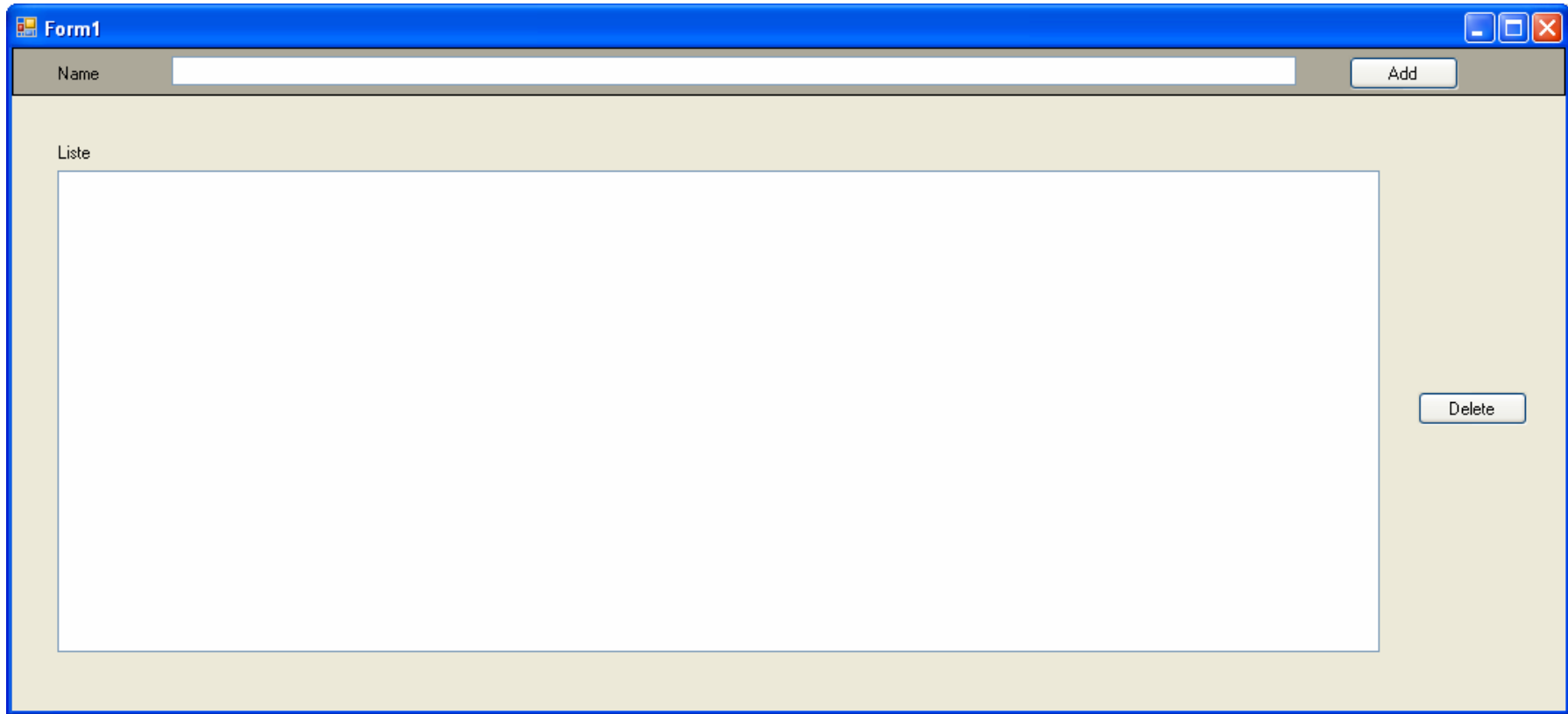
private void BnDelete_Click(object sender, EventArgs e)    {
    int i = Liste.SelectedIndex;
    string sStr = i.ToString();
    MessageBox.Show(sStr);
    if (i >= 0)  {
        DialogResult retCode = MessageBox.Show("Wollen Sie
wirklich den Eintrag löschen?",
        "Remove Item", MessageBoxButtons.YesNo);
        if (retCode == DialogResult.Yes) {
            MessageBox.Show("Löschen");
            Liste.Items.RemoveAt(i);
        } // if
    } // if
} // BnDelete_Click

```


4. Beispiel



4. Beispiel



Ablauf eines Kopiervorgangs:

- Alle GUI-Elemente markieren, STRG+C
- Neues Projekt, Beispiel4
- Einfügen mit STRG+V
- Panel einfügen
 - Eigenschaft „Dock“ auf „Top“ setzen
 - BackColor „ControlDark“
 - BorderStyle „FixedSingle“
 - AutoSize auf „True“ (Optional)
- Label, Textfeld und Schalter markieren, Ausschneiden, Panel anklicken und Einfügen
- Ändern der Anchor-Eigenschaften für Textfeld und Schalter
- Ändern der Anchor-Eigenschaften für die Liste und den Remove-Schalter

Eigenschaften der GUI-Elemente: SplitContainer

- Name
- Anchor: Top, Left (Fullsize)
- BackColor
- BackgroundImage
- BorderStyle
- Enabled
- FixedPanel
- Font
- ForeColor
- IsSplitterFixed
- Location
- Orientation Vertical / Horizontal
- Panel1
- Panel2
- Panel1MinSize
- Panel2MinSize
- Size
- SplitterDistance, linker R.
- SplitterIncrement
- SplitterWidth
- TabIndex
- TabStop
- Tag
- Visible

OpenFileDialog

```
OpenFileDialog openFileDialog1 = new OpenFileDialog();
openFileDialog1.Filter =
    "txt files (*.txt)|*.txt|pdf files (*.pdf)|*.pdf|All files (*.*)|*.*";
openFileDialog1.FilterIndex = 2;// zählt von 1 !
openFileDialog1.DefaultExt = ".txt";
openFileDialog1.InitialDirectory="v:\\daten\\0";
openFileDialog1.Multiselect=false;
openFileDialog1.RestoreDirectory = true;
if (openFileDialog1.ShowDialog() == DialogResult.OK){
    // Verarbeitung
}
```

OpenFileDialog

```
if (openFileDialog1.ShowDialog() == DialogResult.OK){  
    string sFile=openFileDialog1.FileName.ToString()  
}
```

```
if (openFileDialog1.ShowDialog() == DialogResult.OK){  
    string[] sFiles;  
    sFiles = openFileDialog1.FileNames;  
    foreach (string sFile in sFiles)  
    {  
        MessageBox.Show(sFile, Application.ProductName);  
    }  
}
```

SaveFileDialog

```
SaveFileDialog saveFileDialog1 = new SaveFileDialog();
saveFileDialog1.Filter =
    "txt files (*.txt)|*.txt|pdf files (*.pdf)|*.pdf|All files (*.*)|*.*";
saveFileDialog1.FilterIndex = 1;// zählt von 1 !
saveFileDialog1.DefaultExt = ".txt";
saveFileDialog1.InitialDirectory = "c:\\daten";
saveFileDialog1.RestoreDirectory = true;
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    string sFile = saveFileDialog1.FileName.ToString();
    MessageBox.Show(sFile, Application.ProductName);
}
```

Eigenschaften der GUI-Elemente: Frame

- Name
- AutoScroll False/True
- AutoSize False/True
- AutoSizeMode: GrowOnly, GrowAndShrink
- BackColor
- BackGroundImage
- Cursor
- DoubleBuffered
- Enabled
- Font
- ForeColor
- FormBorderStyle:
 - None
 - FixedSingle,
 - Fixed3D,
 - FixedDialog,
 - Sizable,
 - FixedToolWindow,
 - SizableToolWindow
- CancelButton
- MaximizeBox
- MinimizeBox

Eigenschaften der GUI-Elemente: Frame

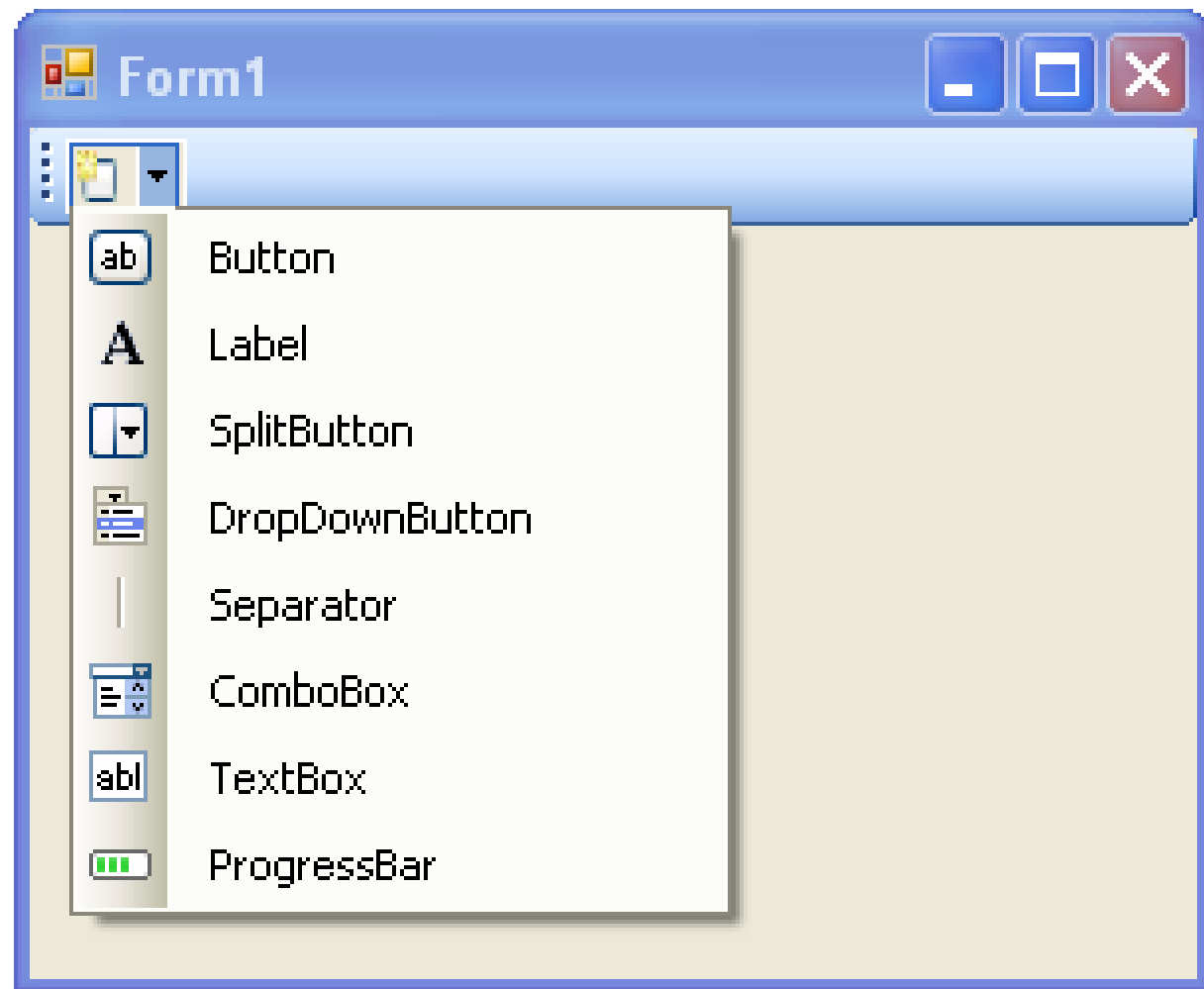
- Icon
- KeyPreview
- Location
- Locked
- MainMenuStrip
- Opacity
- ShowIcon
- Size
- StartPosition:
 - Manual,
 - CenterScreen,
 - WindowsDefaultLocation,
 - WindowsDefaultBounds,
 - CenterParent
- Tag
- Text: (Caption)
- TopMost False/True
- WindowState:
 - Normal,
 - Minimized,
 - Maximized

Eigenschaften der GUI-Elemente: Menüs

- Checked
- Enabled
- Font
 - Name
 - Size
 - Unit (Einheit)
 - Bold
 - Italic
 - Underline
- ForeColor
- Image
- ImageAlign
- ImageScaling
- Tag
- Text
- TooltipText
- Visible

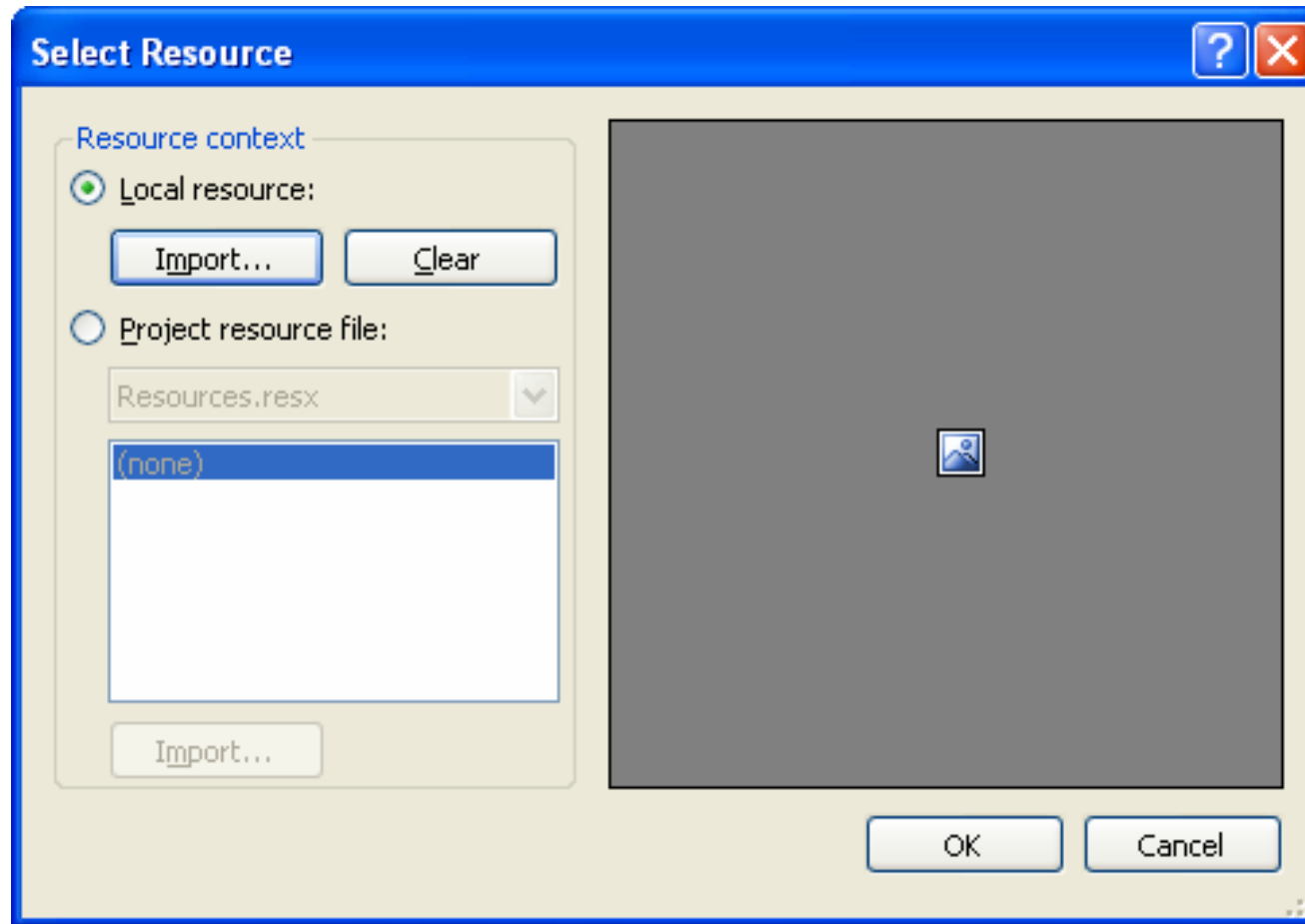
Eigenschaften der GUI-Elemente: Symbolleisten

- Name: **ToolStrip**
- Dock !
 - Top
 - Left
 - Right
 - Bottom



ToolStrip mit einem Schalter

- Laden eines Icons für einen Schalter
- Einfügen, Properties, Eintrag „Image“



ToolStrip

■ SplitButton

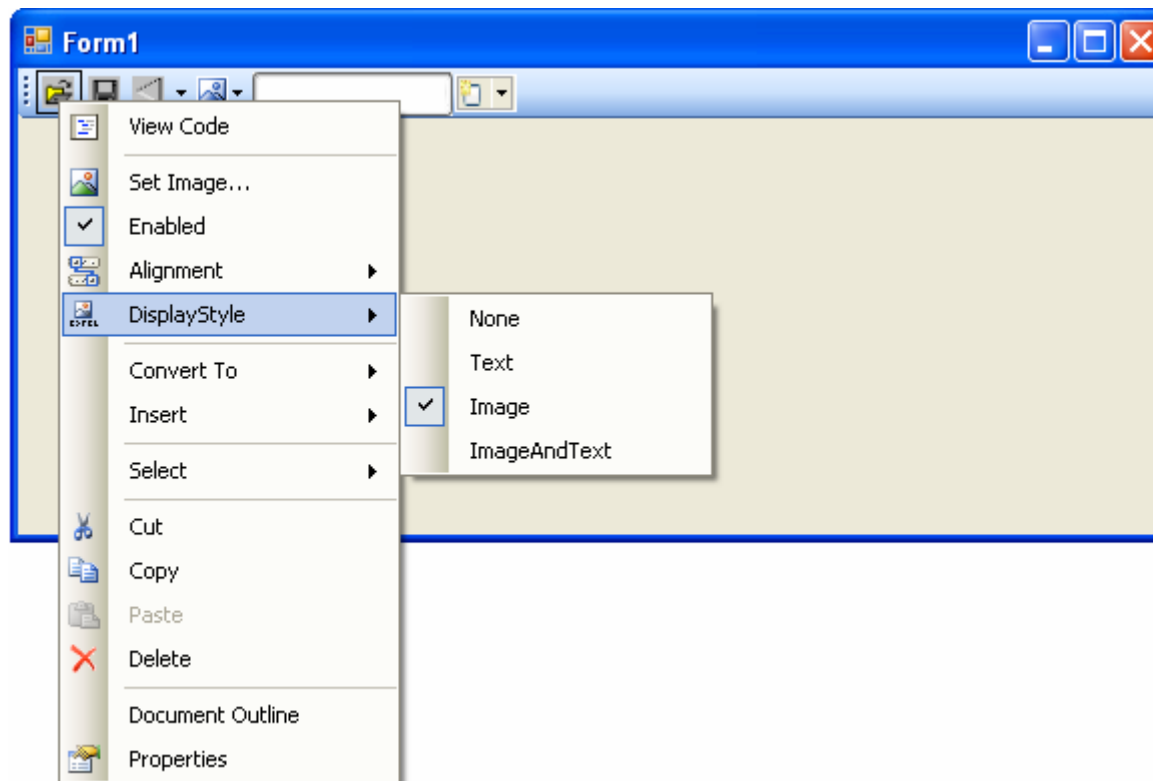
- Ein Splitbutton hat immer einen Schalter sichtbar
- Damit kann man direkt ihn anwählen
- Mit dem Pfeil kann man alle weiteren Einträge sichtbar machen und auswählen

■ DropDownButton

- Ein DropDownButton zeigt **nie** einen Schalter
- Man muss immer den mit dem Pfeil den Eintrag auswählen

ToolStrip

- Bei allen Elementen kann man das Aussehen definieren
 - Rechte Maustaste klicken
 - DisplayStyle



ToolStrip

- Für Schalter kann man CheckOnClick aktivieren
- Damit bleibt dieser gedrückt oder nicht gedrückt
- Den Status kann man im Programm abfragen

