

Web-Entwicklung mit Microsoft Silverlight – Grundlagen und Aktionsprogrammierung

Dr. Thomas Meinike
Hochschule Merseburg (FH)

05.11.2008 – Wiesbaden

Silverlight?

→ Einstieg (1)

- **Silverlight** ist eine RIA-Technologie von Microsoft
- Browser-Plugin für Windows und MacOS X
- Novell arbeitet an Linux-Unterstützung (Moonlight)
- Bisherige Versionen: 1.0¹ (09/2007) und 2 (10/2008)
- Benutzerschnittstelle wird mit der deklarativen XML-Sprache XAML realisiert
- Programmcode für Interaktionen und Datenaustausch wird mit JavaScript (SL1) bzw. C# oder VB.NET (SL2) umgesetzt (weitere Sprachen möglich)
- Technologisch konkurriert Silverlight mit Adobe Flash/Flex und auf der Vektorgrafikebene ansatzweise mit SVG

¹) Hinweis: Microsoft unterscheidet zwischen Silverlight 1.0 und 2 – im Folgenden steht 1 für 1.0

→ Einstieg (2)

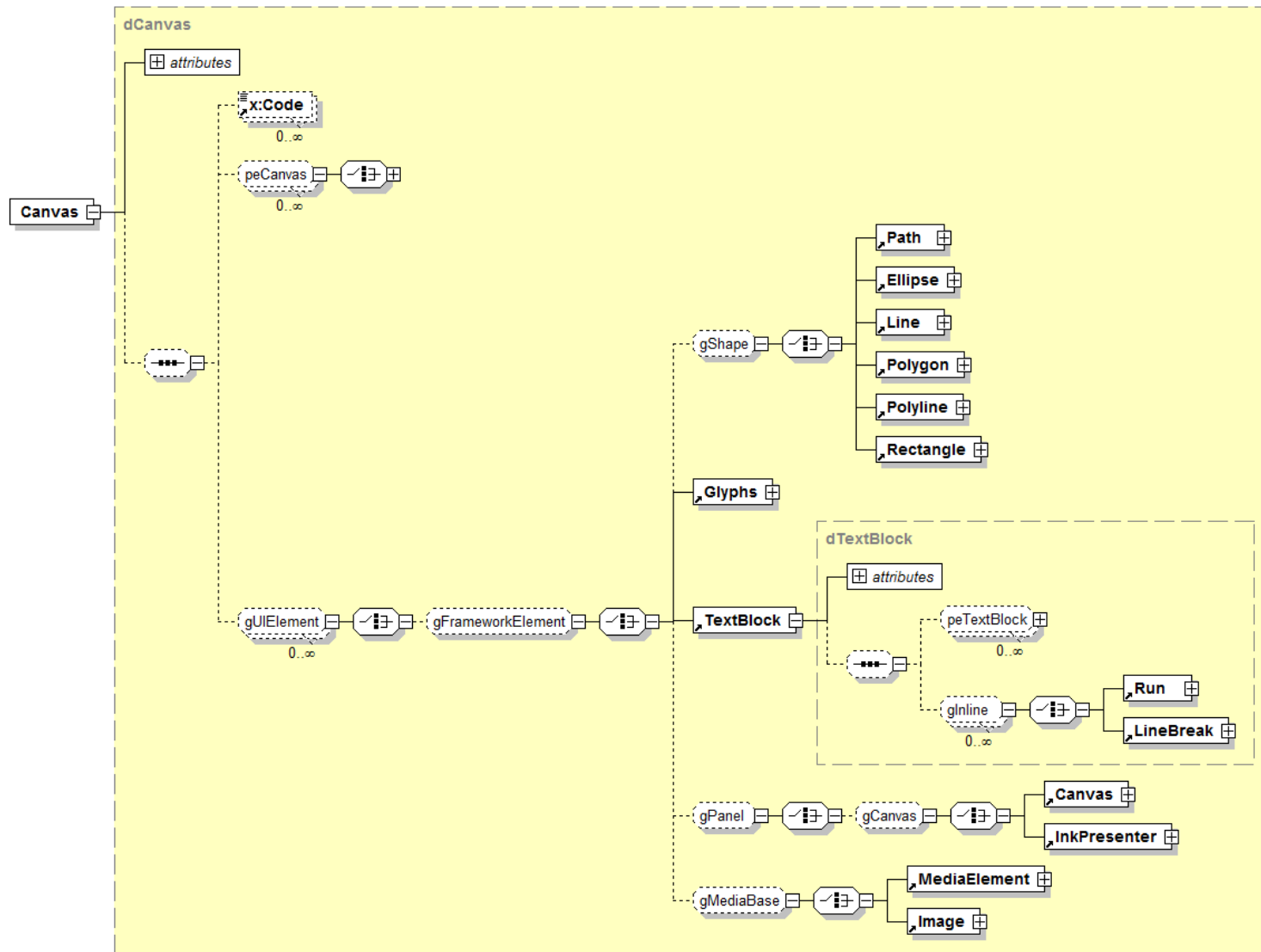
- **Entwicklerzentrale** → <http://silverlight.net>
- **Microsoft-Werkzeuge für Designer:**
 - Microsoft Expression (Blend, Design, Encoder, Media, Web)
- **Microsoft-Werkzeuge für Entwickler:**
 - Visual Studio 2008 SP1 (kommerziell) + SL-Tools
 - Visual Web Developer Express 2008 SP1 (frei) + SL-Tools
- **Sonstige freie Werkzeuge:**
 - XAML: Kaxaml, XamlPadX, Inkscape, ...
 - Code: Dynamic Language Runtime SDK (Kommandozeilentool Chiron.exe sowie DLLs für JScript, IronPython und IronRuby)
- **Dokumentation:**
 - MSDN-Referenz online verfügbar
 - Dokumentation im CHM-Format (52 MB!)

XAML

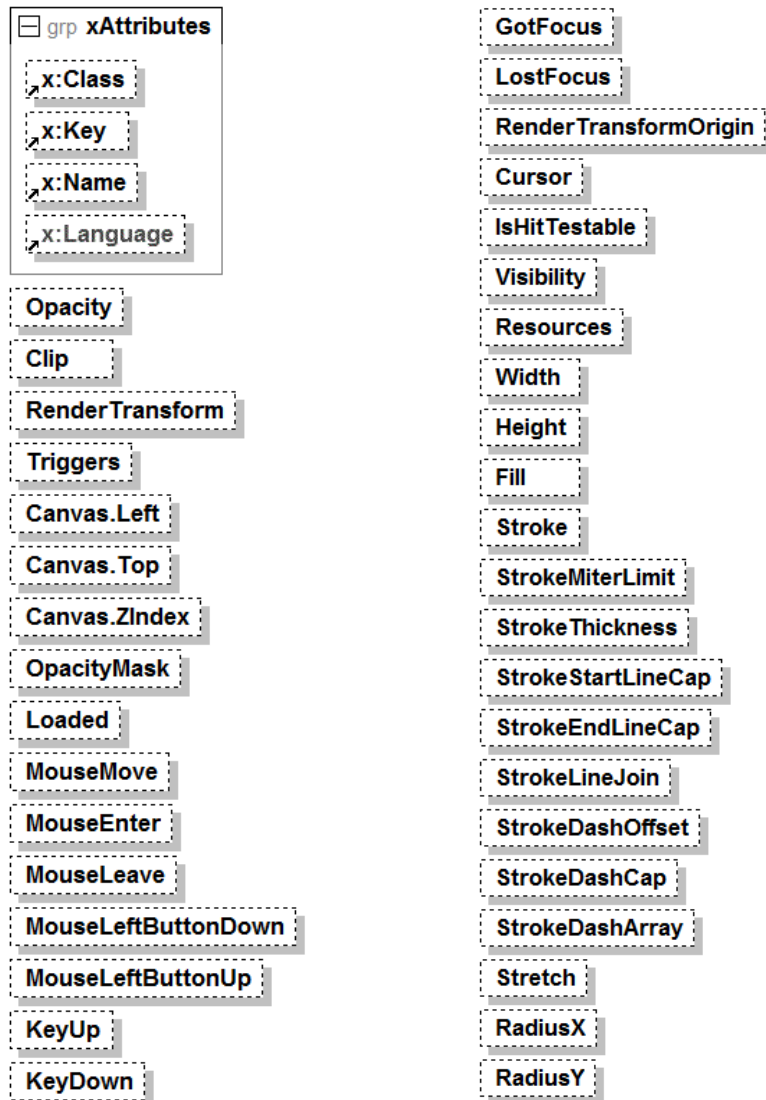
→ Überblick

- **XAML = Extensible Application Markup Language**
- Basiert auf der Windows Presentation Foundation (WPF) des .NET-Frameworks und dient zur Beschreibung von Benutzeroberflächen für Windows und als Untermenge für Silverlight
- XML-Schemata definieren mögliche Elemente und Attribute (SL1: 112 KB vs. SL2: 3.6 MB)
- Enthalten sind Elemente für Zeichenfläche, Vektorformen, Text und speziellere für Animationen, Transformationen, ...
- Attribute zur Vorgabe von Koordinaten und Formatierungen
- Silverlight 2 stellt eine stark erweiterte Palette an Steuerelementen (Controls) zur Verfügung

→ Auszug aus dem XML-Schema für Silverlight 1



→ Auszug aus dem XML-Schema für Silverlight 1



Attribute des Rectangle-Elements

- Definitionen: Width, Height, ...
- Formatierungen: Fill, Stroke, ...
- Event-Handler: MouseEnter, ...
- Identifikatoren: x:Key, x:Name, ...

→ Grundgerüst eines XAML-Dokuments

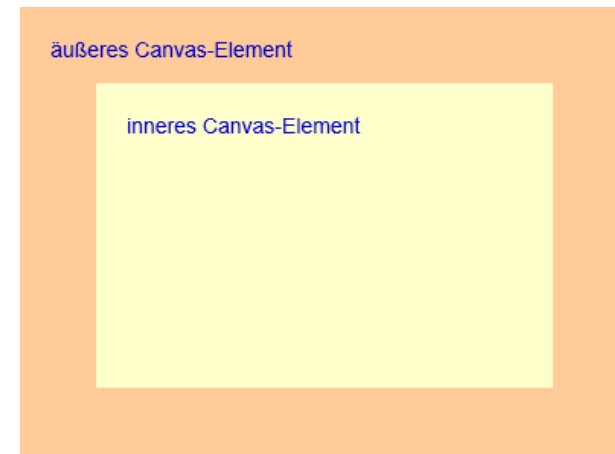
```

<?xml version="1.0" encoding="UTF-8"?>
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Background="#FC9" Width="400" Height="300">
    <Canvas Canvas.Top="50" Canvas.Left="50" Width="300" Height="200" Background="#FFC">
        <!-- inneres Canvas-Element -->
    </Canvas>
<!-- äußeres Canvas-Element -->
</Canvas>

```

←----- SL1-Namensraum
und x-NS für Namen

- Canvas legt die Zeichenfläche fest und kann weitere Canvas-Kinderelemente enthalten
- Unterhalb von Canvas werden die Vektorformen wie Rechteck, Ellipse, Linien, Pfade sowie Textinhalte deklariert
- Canvas.Left / Canvas.Top beschreiben die x- bzw. y-Richtung im Bezug zum Canvas-Element



→ Vektorform Rechteck (Element Rectangle)

```
<Rectangle Canvas.Left="20" Canvas.Top="20" Width="200" Height="100"  
          Fill="#EEE" Stroke="#000"/>
```



```
<Rectangle Canvas.Left="250" Canvas.Top="20" Width="200" Height="100"  
          Fill="#FFC" Stroke="#00C" RadiusX="10" RadiusY="10"/>
```



- Festlegung der Koordinaten für die linke obere Ecke sowie der Breite und Höhe
- Angaben zur Füll- und Rahmenfarbe
- RadiusX bzw. RadiusY ermöglichen abgerundete Ecken

Hinweis:
Lage und Größe der dargestellten
Objekte wurden willkürlich gewählt!

→ Vektorform Ellipse (Element Ellipse)

```
<Ellipse Canvas.Left="20" Canvas.Top="50" Width="200" Height="100"  
  Fill="#EEE" Stroke="#F00" StrokeThickness="5"/>
```



```
<Ellipse Canvas.Left="250" Canvas.Top="30" Width="75" Height="150"  
  Fill="#FF0" Stroke="#00C" StrokeDashArray="5,2"/>
```



```
<Ellipse Canvas.Left="350" Canvas.Top="50" Width="100" Height="100"  
  Fill="#9CC" Stroke="#090" StrokeThickness="2"/>
```



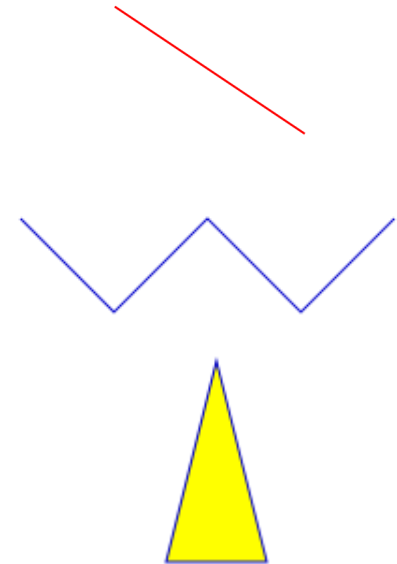
- Festlegung der Koordinaten sowie der Breite und Höhe (Achsendurchmesser)
- Angaben zur Rahmenformatierung (Linienfarbe, -form und -stärke, analog bei anderen Vektorformen)
- Kein separates Element für Kreise – ein Kreis ergibt sich durch Height = Width

→ Vektorformen mit Linien (Elemente Line, Polyline, Polygon)

```
<Line X1="30" Y1="20" X2="300" Y2="200" Stroke="#F00"/>
```

```
<Polyline Points="25,75 75,125 125,75 175,125 225,75" Stroke="#00C"/>
```

```
<Polygon Points="150,150 200,150 175,50" Fill="#FF0" Stroke="#00C"/>
```

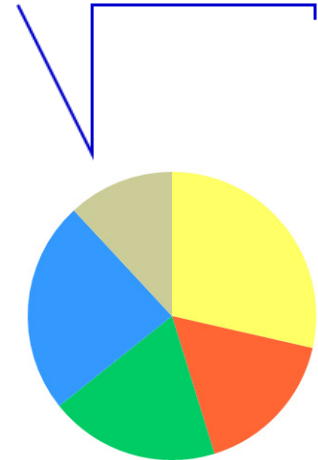


- Einfache Linien werden durch zwei Punkte definiert
- Polygon- und Polyline-Elemente erhalten Punktliste im Points-Attribut
- Bei Polygon-Elementen werden Start- und Endpunkt automatisch verbunden
- Weitere Angaben zur Linienformatierung möglich, siehe Element Ellipse

→ Vektorformen mit Pfaden (Element Path)

```
<Path Data="M100,100 L150,200 V100 H300 V110" Stroke="#00C"/>
```

```
<Path Data="M200,200 L82.73,293.52 A150,150 0 0,1 97.97,90.04 Z"
Fill="#39F"/><!-- blaues Segment im Kreisdiagramm -->
```



- Pfade ermöglichen komplexere Formen: Linien, Bögen, Bézier-Kurven
- Pfad-Daten werden beim Data-Attribut angegeben
[Bezeichner wie u. a. M(ove), L(ine), V(ertical), H(orizontal), A(rc), Z(ClosePath)
in Analogie zu SVG verwendbar]

→ Texteinbindung (Element TextBlock)

```
<TextBlock Text="Hallo Silverlight!" Canvas.Left="30" Canvas.Top="30"  
    FontFamily="Arial" FontSize="36" Foreground="#00C"/>
```

Hallo Silverlight!

```
<TextBlock Canvas.Left="20" Canvas.Top="20" FontFamily="Arial" FontSize="16"  
    TextWrapping="Wrap" Width="300">Dieser Text wird über mehrere Zeilen verteilt. Mit  
    Run lassen sich <Run Foreground="#F00" Text="Textteile formatieren"/>.<LineBreak/>  
    Zeilenumbrüche können mit LineBreak erzeugt werden.</TextBlock>
```

Dieser Text wird über mehrere Zeilen verteilt. Mit Run lassen sich **Textteile formatieren**. Zeilenumbrüche können mit LineBreak erzeugt werden.

- TextBlock-Elemente können Textinhalte einzeilig oder mehrzeilig darstellen
- Textformatierung mit Attributen (u. a. für Schriftart, -farbe und -größe)
- Kindelemente Run zur Auszeichnung von Textteilen sowie LineBreak für Umbrüche

→ Rasterbilder, Audio, Video (Elemente Image und MediaElement)

```
<Image Source="bild.jpg" Width="..." Height="..." Canvas.Left="..." Canvas.Top="..." />  
  
<MediaElement x:Name="audio" Source="audio.mp3" AutoPlay="True" />  
  
<MediaElement Source="video.wmv" Width="..." Height="..." AutoPlay="False"  
    Canvas.Left="..." Canvas.Top="..." />
```

- Rasterbildformate: JPEG, PNG
- Audio-Formate: MP3, WMA (künftig auch AAC)
- Video-Formate: WMV 7-9 (inkl. HD 720p, künftig auch H.264/MPEG-4) sowie Streaming
- Steuerung der Audio- und Video-Wiedergabe über Scripting-Methoden play(), pause(), stop()
- Bearbeitung von Material mit Expression Media / Encoder

→ Farbverläufe (Elemente LinearGradientBrush und RadialGradientBrush)

```
<Rectangle Width="200" Height="100" Canvas.Left="20" Canvas.Top="20">
  <Rectangle.Fill>
    <LinearGradientBrush StartPoint="0,0" EndPoint="1,1">
      <GradientStop Offset="0.0" Color="#CCC"/>
      <GradientStop Offset="1.0" Color="#090"/>
    </LinearGradientBrush>
  </Rectangle.Fill>
</Rectangle>
```



```
<Rectangle Width="200" Height="100" Canvas.Left="20" Canvas.Top="20">
  <Rectangle.Fill>
    <RadialGradientBrush>
      <GradientStop Offset="0.0" Color="#FFF"/>
      <GradientStop Offset="1.0" Color="#090"/>
    </RadialGradientBrush>
  </Rectangle.Fill>
</Rectangle>
```

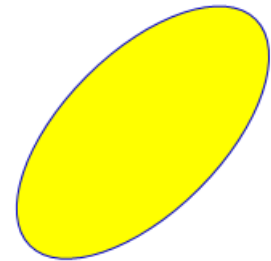


- Brushes dienen als spezielle Füllungen
(ImageBrush, LinearGradientBrush, RadialGradientBrush, SolidColorBrush, VideoBrush)
- SVG-ähnliche Deklaration von Farbverläufen

→ Koordinatensystem-Transformationen

```
<Ellipse Canvas.Left="20" Canvas.Top="20" Width="100" Height="200" Stroke="#00C" Fill="#FF0">  
  <Ellipse.RenderTransform>  
    <RotateTransform Angle="45" CenterX="70" CenterY="120"/>  
  </Ellipse.RenderTransform>  
</Ellipse>
```

```
<Ellipse Canvas.Left="20" Canvas.Top="20" Width="100" Height="200" Stroke="#00C" Fill="#FF0">  
  <Ellipse.RenderTransform>  
    <TransformGroup>  
      <RotateTransform Angle="45" CenterX="70" CenterY="120"/>  
      <TranslateTransform X="100" Y="50"/>  
    </TransformGroup>  
  </Ellipse.RenderTransform>  
</Ellipse>
```



- Transformationsarten: Rotate, Scale, Skew, Translate und Matrix
- Mehrere Transformationen über TransformGroup kombinierbar

→ Animationen (Beispiel Bewegung eines Rechtecks)

```

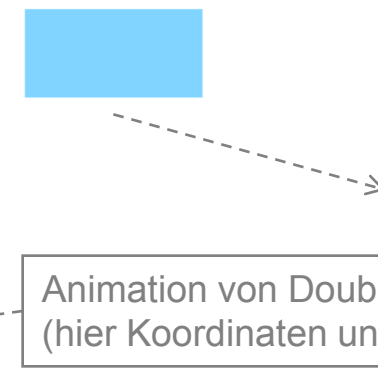
<Canvas ...>

  <Rectangle x:Name="rechteck" Width="200" Height="100"
    Canvas.Left="0" Canvas.Top="0" Fill="#6CF"/>

  <Canvas.Triggers>
    <EventTrigger RoutedEvent="Canvas.Loaded">
      <EventTrigger.Actions>
        <BeginStoryboard>
          <Storyboard>
            <DoubleAnimation ←
              Storyboard.TargetName="rechteck"
              Storyboard.TargetProperty="(Canvas.Left)" To="250" Duration="00:00:05"/>
            <DoubleAnimation
              Storyboard.TargetName="rechteck"
              Storyboard.TargetProperty="(Canvas.Top)" To="250" Duration="00:00:05"/>
            <DoubleAnimation
              Storyboard.TargetName="rechteck"
              Storyboard.TargetProperty="Opacity" To="0.2" Duration="00:00:05"/>
          </Storyboard>
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
  </Canvas.Triggers>

</Canvas>

```



Animation von Double-Werten
(hier Koordinaten und Opacity)

→ Zuweisung von Farbwerten (Beispiel Farbe **Rot**)

- **#FF0000** (#RRGGBB | hexadezimal 6-stellig)
- **#F00** (#RGB | hexadezimal 3-stellig, Einzelwerte werden verdoppelt)
- **#FFFFFF0000** (#AARRGGBB | hexadezimal 8-stellig)
- **#FF00** (#ARGB | hexadezimal 4-stellig, Einzelwerte werden verdoppelt)
- **sc#1,0,0** (sc#r,g,b | Farbwerte 0...1)
- **sc#1,1,0,0** (sc#a,r,g,b | Farbwerte 0...1)
- **Red** (Farbwort | 141 vordefiniert)

➤ Hinweise:

A/a = Alpha-Kanal, R = Rot, G = Grün, B = Blau

Typische Anwendung bei den Attributen Background, Fill, Foreground, Stroke, ...

Farben im Code dynamisch zuweisen: `Color.FromArgb(A, R, G, B)` [jeweils 0...255]

Einbindung in HTML

Einbindung in HTML

→ Silverlight 1 – Einbindung über silverlight.js aus dem SDK

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="de" lang="de">

<head>
  <title>Silverlight 1 in HTML</title>
  <script src="silverlight.js" type="text/javascript"></script>
</head>

<body>
  <div id="SilverlightPlugInHost">
    <script type="text/javascript">
      createSilverlight();
    </script>
  </div>
  <!-- weitere HTML-Inhalte -->
</body>

</html>
```

```
function createSilverlight()
{
  Silverlight.createObjectEx({
    source: 'filename.xaml',
    parentElement:
      document.getElementById(
        'SilverlightPlugInHost'),
    id: 'SilverlightPlugIn',
    properties: {
      width: '800px',
      height: '600px',
      background: '#ffffff',
      isWindowless: 'false',
      version: '1.0'
    },
    events: {
      onLoad: null,
      onError: null
    }
  });
}
```

- Im (X)HTML-Dokument ist die Bibliothek silverlight.js einzubinden
- Die zusätzlich benötigte JS-Funktion createSilverlight() initialisiert das Plugin
- Sollte das Plugin nicht erkannt werden, erscheint eine Meldung mit Download-Link:
- Zusätzlicher Aktionscode kann in weiteren JS-Dateien liegen



→ Silverlight 2 – Direkte object-Einbindung der XAP-Kompilate

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">

  <head>
    <title>Silverlight 2 in HTML</title>
    <script src="extern.js" type="text/javascript"></script>
  </head>

  <body>
    <object id="SilverlightPlugIn" width="800" height="600"
      type="application/x-silverlight-2" data="data:application/x-silverlight,">

      <param name="background" value="#ffffff" />
      <param name="source" value="filename.xap" />
      <param name="onerror" value="onSilverlightError" />

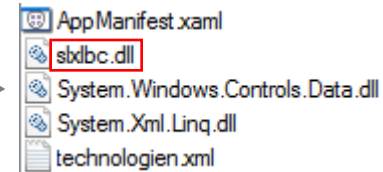
      <a href="http://go.microsoft.com/fwlink/?LinkId=115261">
        
      </a>
    </object>
    <!-- weitere HTML-Inhalte -->
  </body>
</html>
```

- In Visual Studio bzw. VWD Express erstellte XAP-Dateien referenzieren
- JS-Funktion onSilverlightError() für Fehlermeldungen kann in extern.js liegen

Silverlight 2

→ Silverlight 2 – Neuerungen

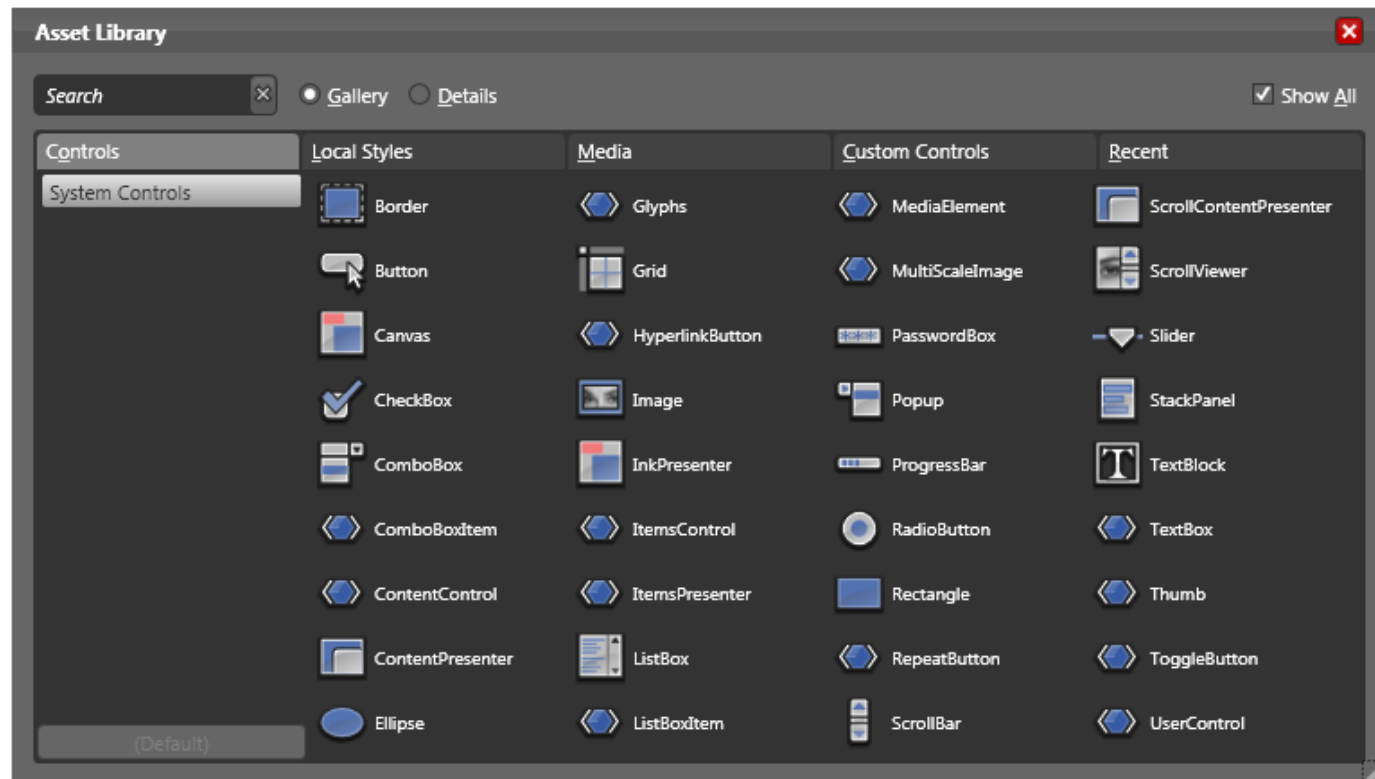
- Ausgeliefert werden XAP-Dateien mit dem kompilierten Projekt in DLL-Form und weiteren Ressourcen; XAP = ZIP-Archiv! ----->



- Größere Auswahl an vordefinierten XAML-Steuerelementen (Controls):

SL1-Basiselemente:


- <> Canvas
- <> Ellipse
- <> Glyphs
- <> Image
- <> InkPresenter
- <> Line
- <> MediaElement
- <> Path
- <> Polygon
- <> Polyline
- <> Rectangle
- <> TextBlock



→ Silverlight 2 – Neuerungen

- Elemente **Button**¹, **CheckBox**², **RadioButton**³ sowie **TextBox**⁴ für Formularanwendungen
- **ComboBox**, **ListBox**⁵, **DataGrid**⁶ verbessern den Umgang mit Daten
- **Calendar**⁷ und **DatePicker** erleichtern die Abfrage von Datumsangaben
- **Border**, **ScrollViewer**⁸, **Slider**, **TabControl**⁹ bereichern die Gestaltung von Oberflächen
- Flexiblere Layouts mit den Steuerelementen **Grid** und **StackPanel**¹⁰ (bisher nur **Canvas**)

1



2

Checkbox 1 Checkbox 2 Checkbox 3

3

Radio Button 1 Radio Button 2 Radio Button 3

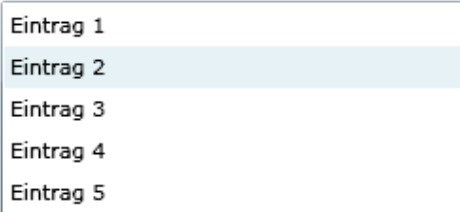
Vorname: 4

Nachname:

8

TextBlock-Element, welches sich wiederum innerhalb eines ScrollViewer-Elements befindet.


5



6

	x	y = x ²	y = x ³
▶	1	1	1
	2	4	8
	3	9	27
	4	16	64
	5	25	125
	6	36	216
	7	49	343
	8	64	512
	9	81	729
	10	100	1000

7



9



→ Silverlight 2 – Neuerungen

➤ StackPanel-Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<UserControl x:Class="slx1bc.Page"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" ←----- SL2-Namensraum
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Width="800" Height="600">

  <Canvas>

    <StackPanel Canvas.Left="30" Canvas.Top="30" Background="#6FF">

      <StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">
        <TextBlock Margin="5" Text="Vorname:" FontSize="14"/>
        <TextBox Height="25" Width="200" FontSize="14"/>
      </StackPanel>

      <StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">
        <TextBlock Margin="5" Text="Nachname:" FontSize="14"/>
        <TextBox Height="25" Width="200" FontSize="14"/>
      </StackPanel>

    </StackPanel>

  </Canvas>
</UserControl>
```



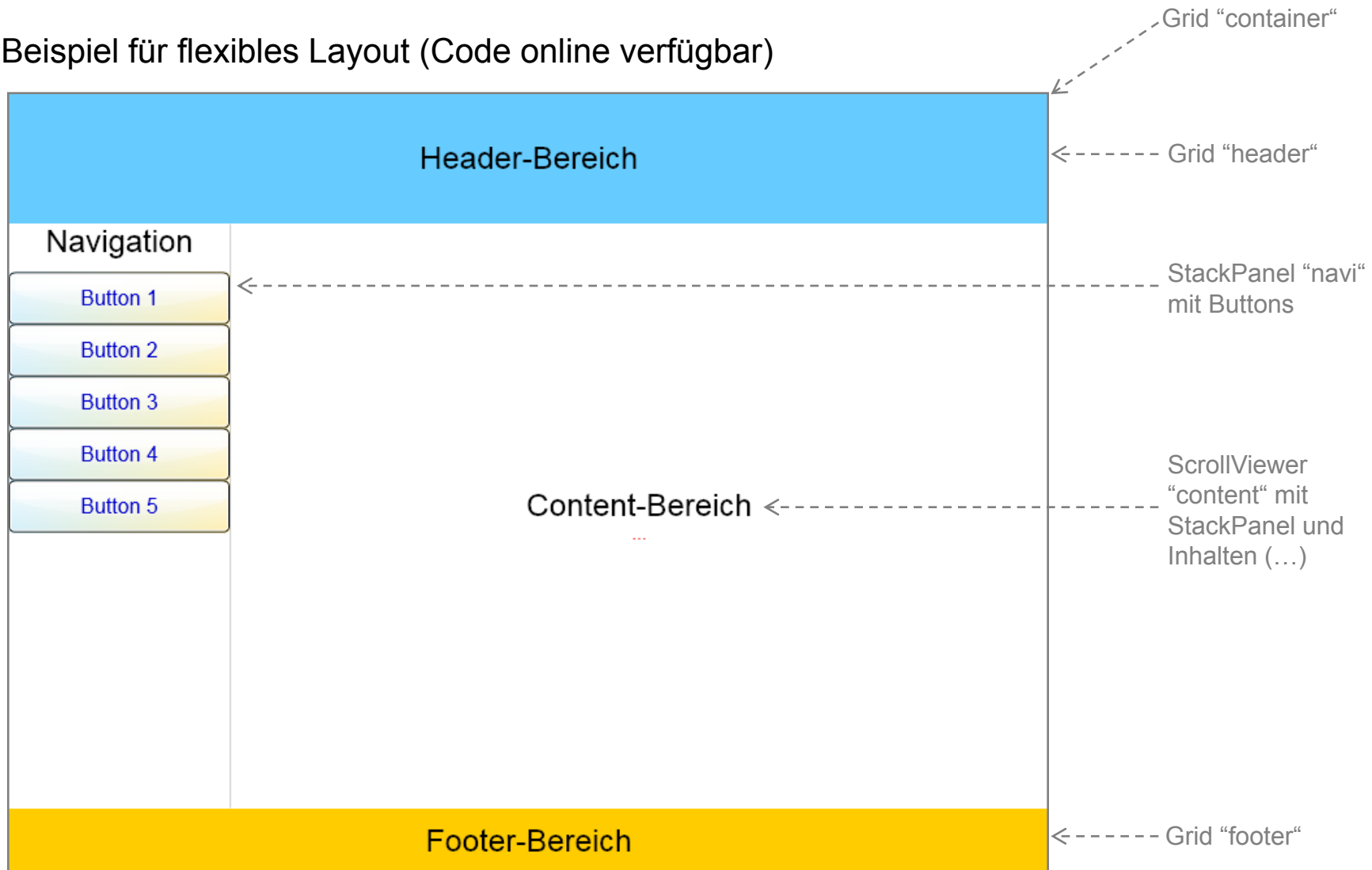
Vorname:

Nachname:

➤ Verschachtelte StackPanel zur relativen Anordnung von untergeordneten Inhalten

→ Silverlight 2 – Neuerungen

- Beispiel für flexibles Layout (Code online verfügbar)



→ Silverlight 2 – Neuerungen

- (A)synchrone Datenkommunikation (HTTP-Dienste, **JSON**, REST, SOAP, XML)
- Klassen WebRequest und WebClient (in SL1 nur über Downloader-Objekt)
- Dynamische Datenbindung (z. B. Bindung von Serverdaten an ein DataGrid)
- Kompakteres Styling von Inhalten über (Style-Element / Style-Attribut)
- Wiederverwendbare eigene Steuerelemente (UserControls)
- XML-Verarbeitung mit LINQ-Techniken
- Cross-Domain-Zugriff auf Ressourcen (erfordert angepasste Policy-Files)
- Verbesserter Silverlight-HTML-Zugriff
- Isolated Storage zur lokalen Datenhaltung
- Dynamic Language Runtime für weitere Sprachen

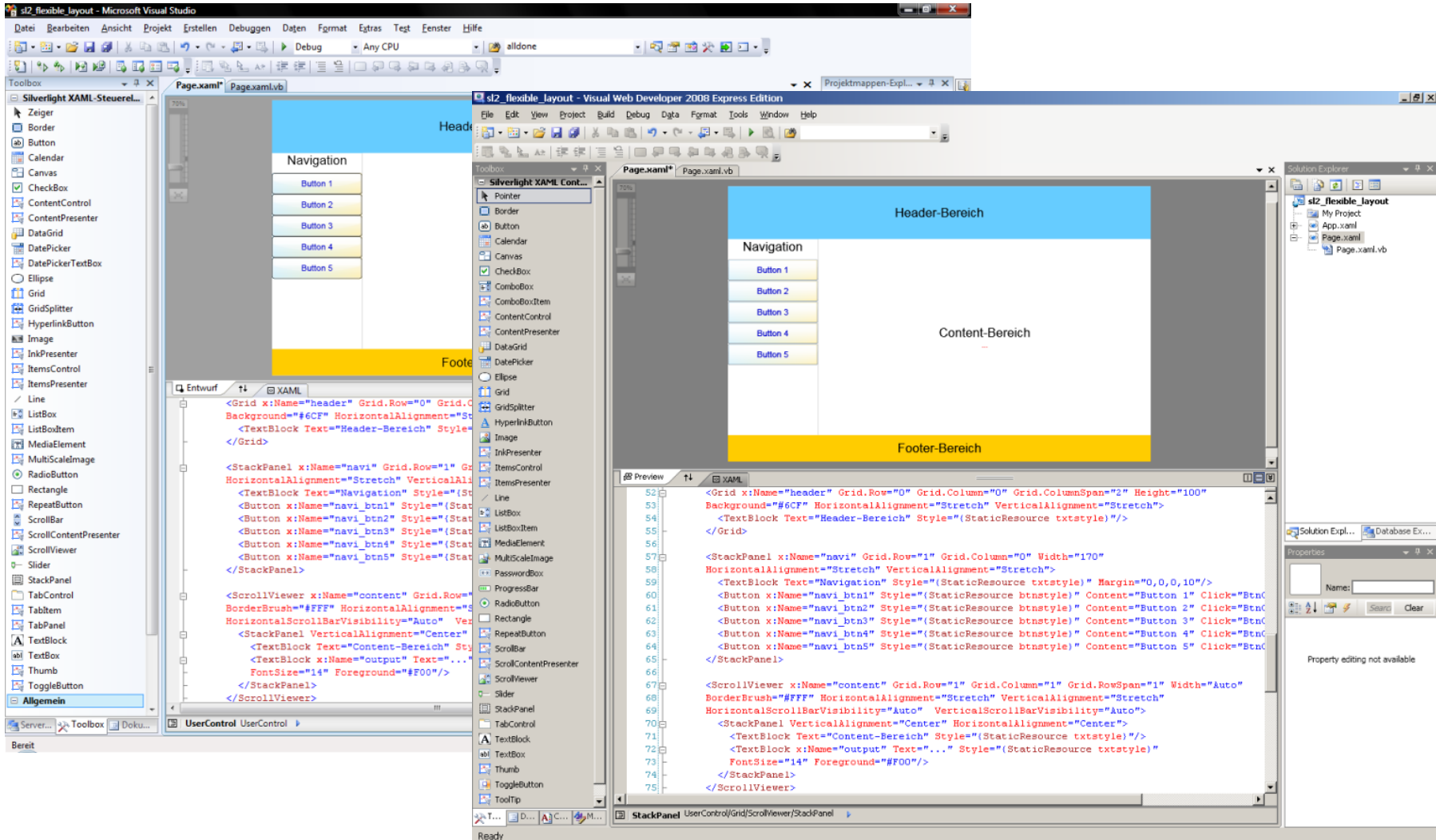
(Siehe zugehörige Codebeispiele)

Olympia 2008-Medaillenspiegel mittels JSON-Daten importiert (2008-08-24)

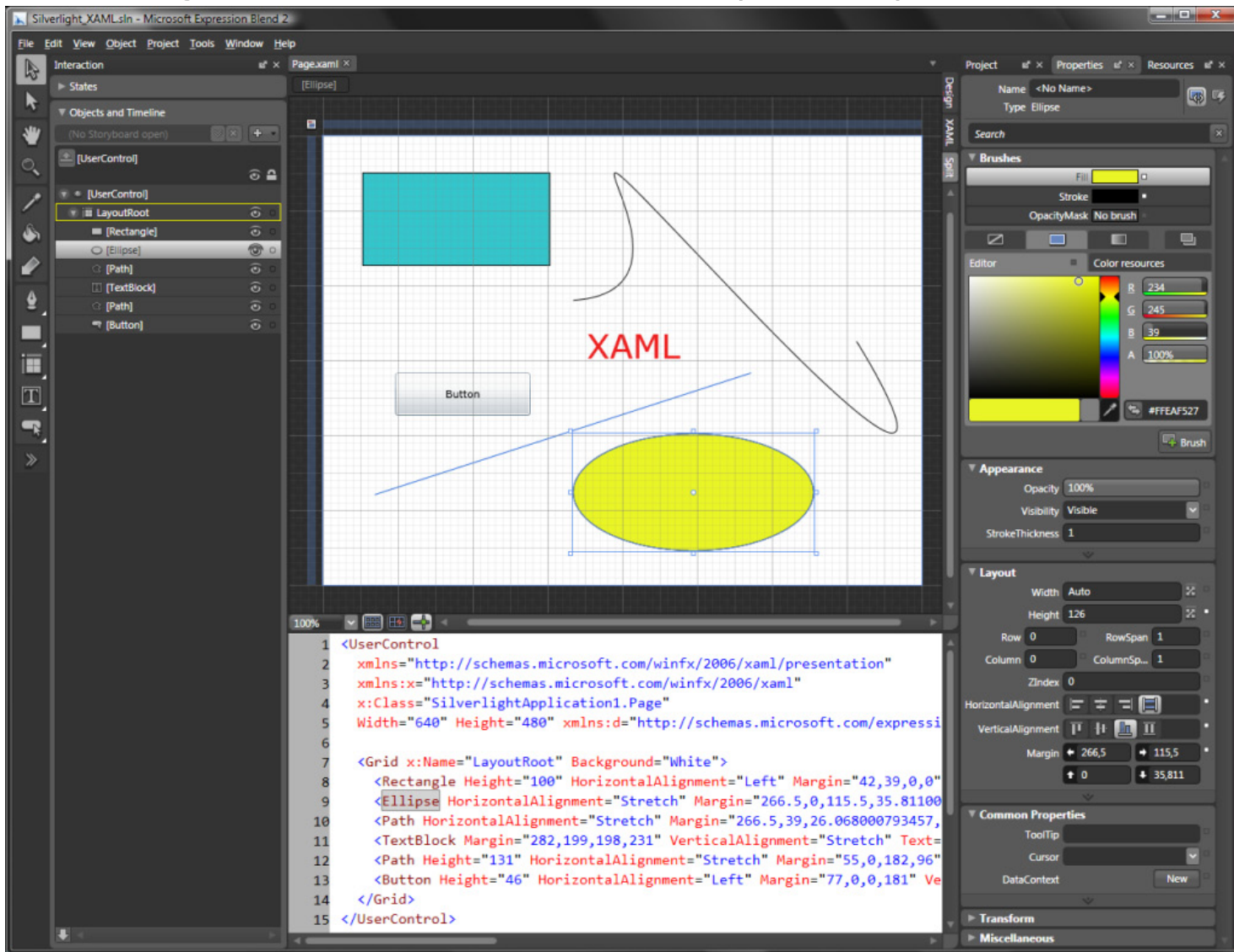
Platz	Land	Gold	Silber	Bronze	Anzahl
1	China	51	21	28	100
2	USA	36	38	36	110
3	Russland	23	21	28	72
4	Großbritannien	19	13	15	47
5	Deutschland	16	10	15	41
6	Australien	14	15	17	46
7	Südkorea	13	10	8	31
8	Japan	9	6	10	25
9	Italien	8	10	10	28
10	Frankreich	7	16	17	40
11	Ukraine	7	5	15	27
12	Niederlande	7	5	4	16
13	Jamaika	6	3	2	11
14	Spanien	5	10	3	18
15	Kenia	5	5	4	14

Werkzeuge

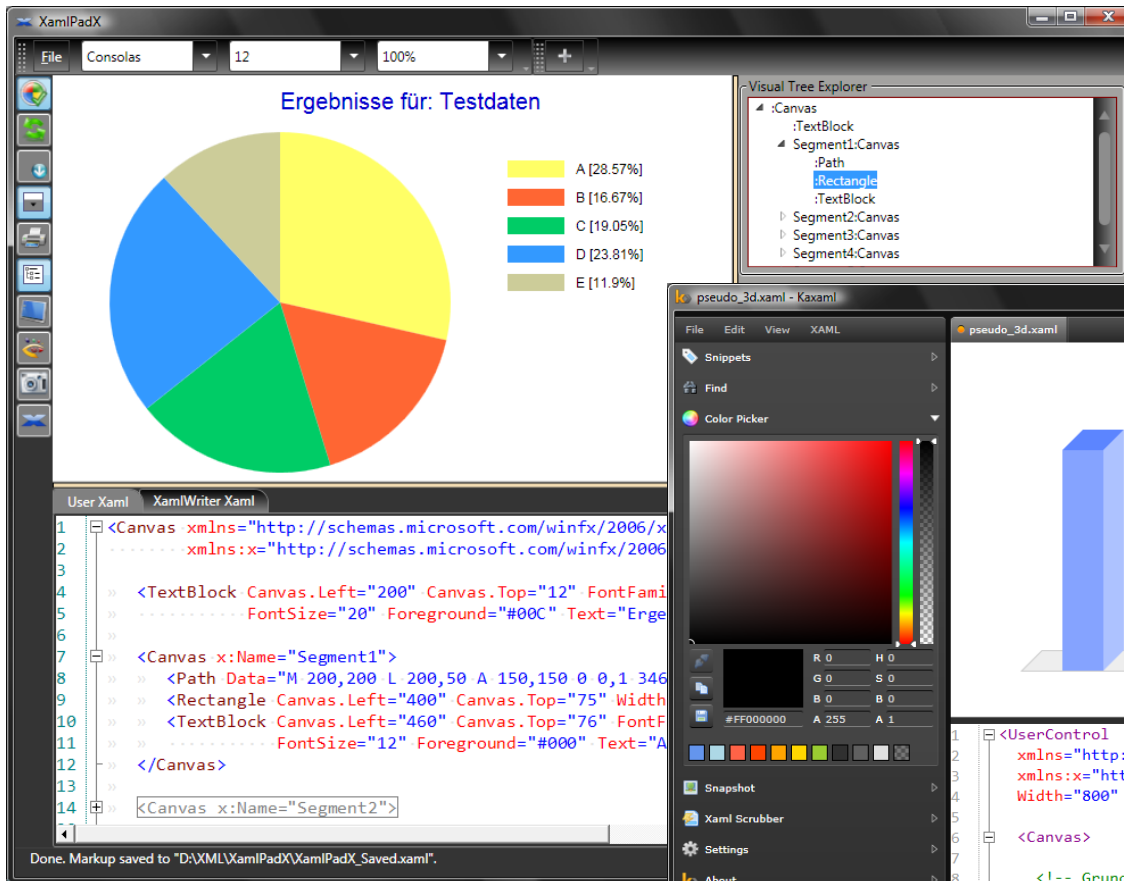
→ Entwicklungsumgebungen Visual Studio 2008 (komm.)
und Visual Web Developer 2008 Express (frei) [<http://microsoft.com>]



→ XAML-Designer Expression Blend 2 (komm.) [<http://microsoft.com>]



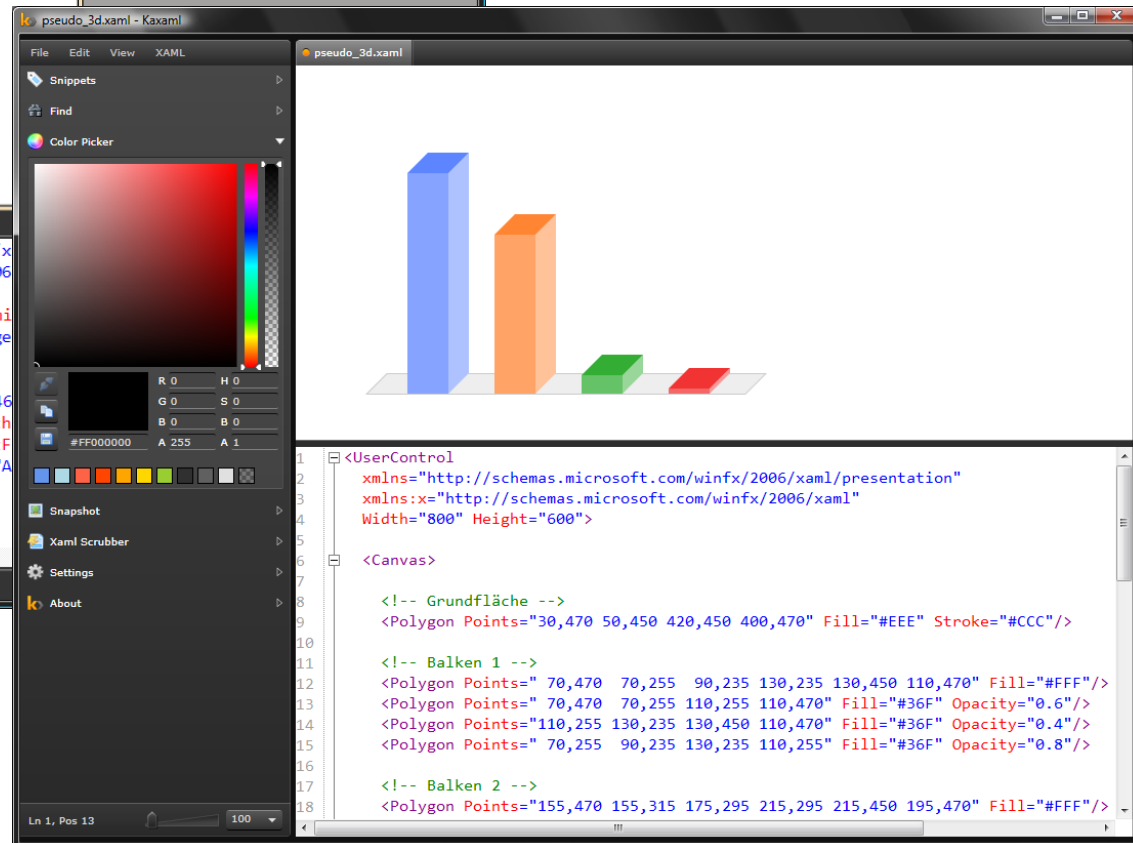
→ XAML-Editoren (frei)



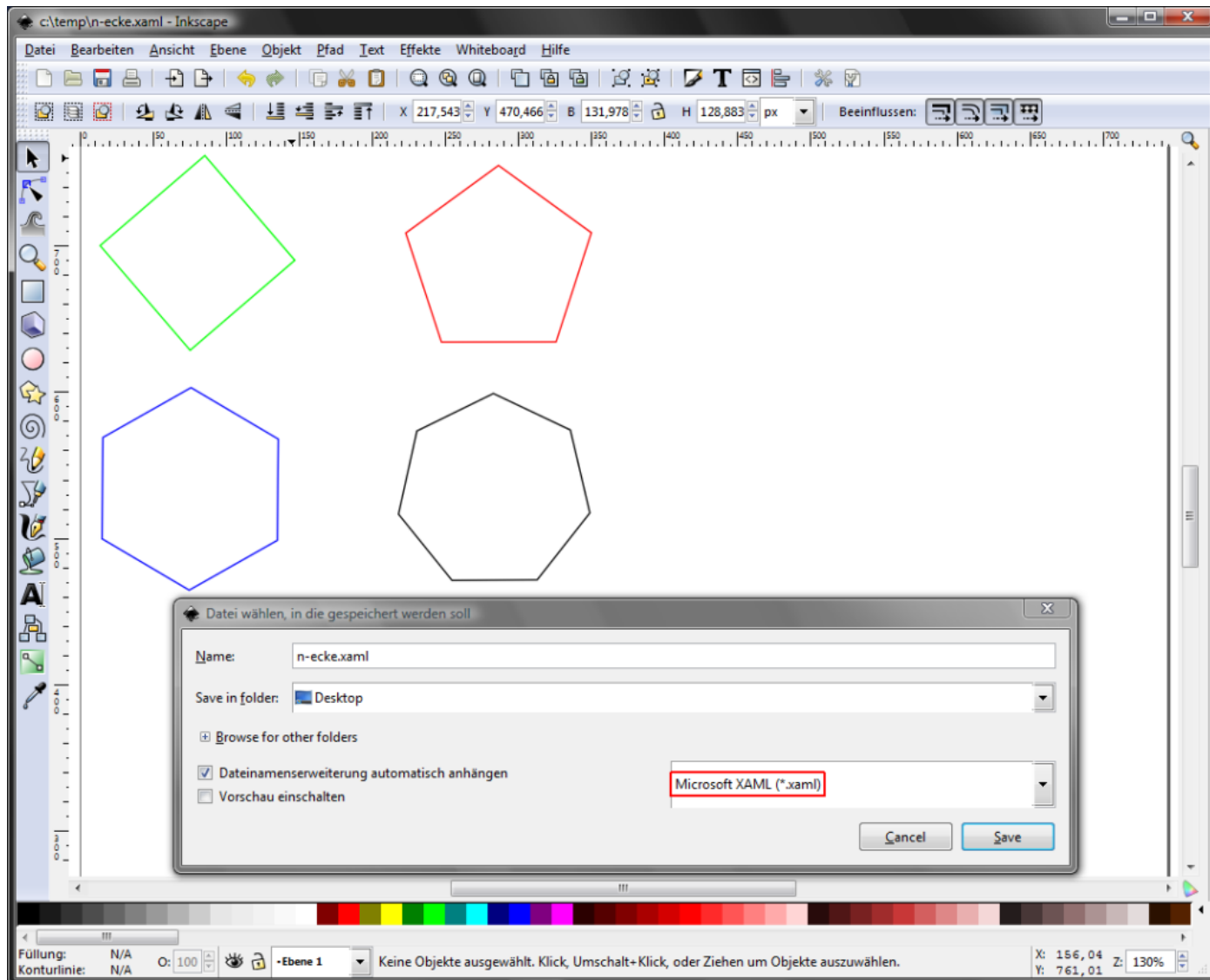
XamlPadX 4.0

[<http://blogs.msdn.com/llobo>]

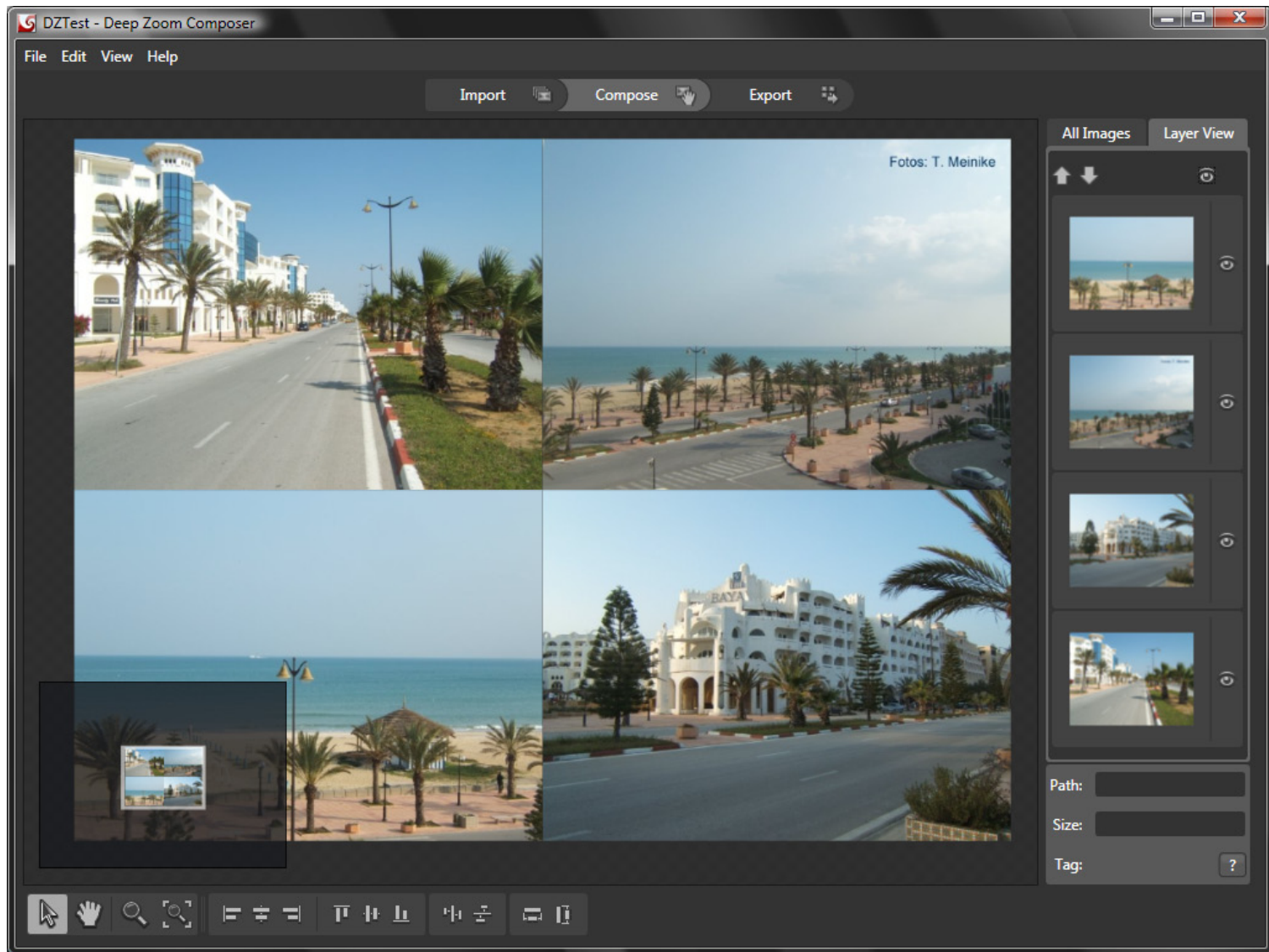
Kaxaml 1.6
[<http://kaxaml.com>]



→ XAML-Export aus SVG-Tool Inkscape (frei) [<http://inkscape.org>]



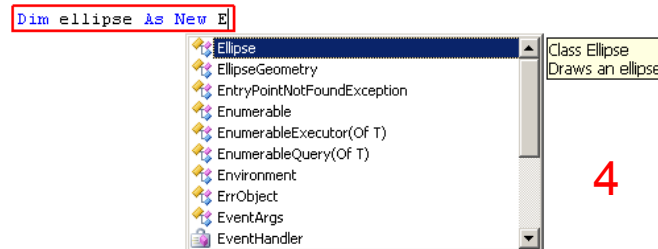
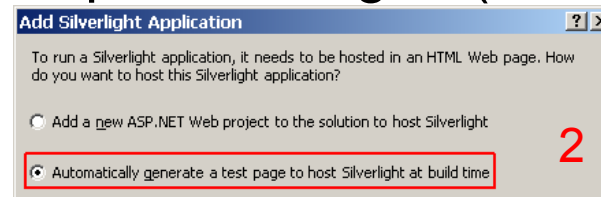
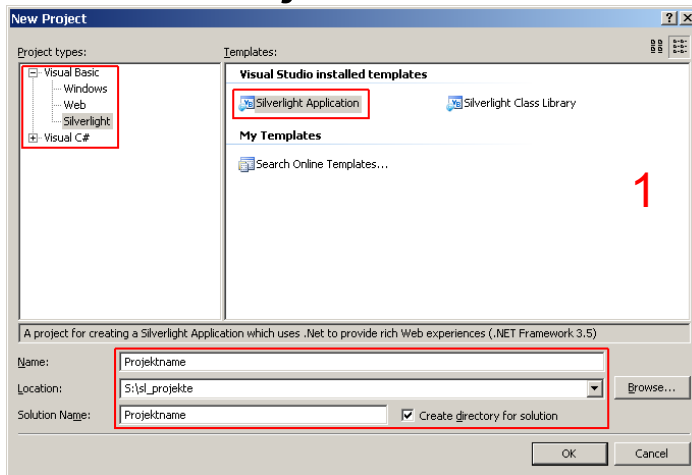
→ Deep Zoom Composer (frei) [<http://microsoft.com>]



Programmierung

Programmierung

→ SL2-Projekt in VWD 2008 Express anlegen (analog in VS 2008)

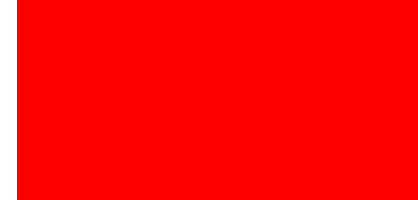


SCHRITTE:

- (1) Neues Projekt anlegen und Sprache wählen (VB.NET oder C#)
- (2) Testserver oder -seite vorgeben
- (3) In der IDE XAML- und Programm-Code schreiben
- (4) „IntelliSense“-Unterstützung nutzen
- Projekt kompilieren und testen (F5)

→ Event-basierte Aktionen im „CodeBehind“ behandeln (1) [VB.NET]

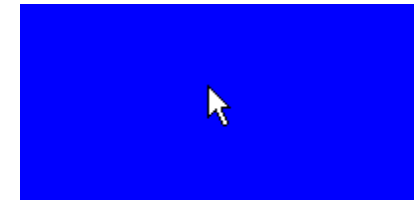
```
<Canvas>
  <StackPanel Orientation="Vertical">
    <Rectangle x:Name="rect" Fill="Red" Width="200" Height="100"
      MouseEnter="NewColor" MouseLeave="OldColor"/>
    <Button Content="Aktion" Width="100" Height="25" Margin="20"
      Click="ChangeRect"/><!-- Interaktionen über Ereignisse -->
  </StackPanel>
</Canvas>
```



```
Private Sub NewColor(ByVal sender As System.Object, _
  ByVal e As System.Windows.Input.MouseEventArgs)
  sender.Fill = New SolidColorBrush(Colors.Blue)
End Sub

Private Sub OldColor(ByVal sender As System.Object, _
  ByVal e As System.Windows.Input.MouseEventArgs)
  sender.Fill = New SolidColorBrush(Colors.Red)
End Sub

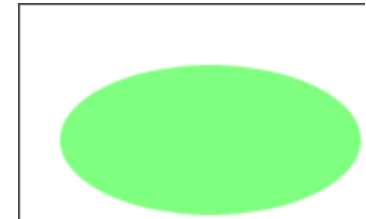
Private Sub ChangeRect(ByVal sender As System.Object, _
  ByVal e As System.Windows.RoutedEventArgs)
  rect.Width = 300
  rect.Fill = New SolidColorBrush(Colors.Yellow)
End Sub
```



→ Event-basierte Aktionen im „CodeBehind“ behandeln (2) [VB.NET]

```
<Canvas Loaded="CreateEllipse"/><!-- Ellipse direkt nach dem Laden erzeugen -->
```

```
Private Sub CreateEllipse(ByVal sender As System.Object, _  
                          ByVal e As System.Windows.RoutedEventArgs)  
    Dim ellipse As New Ellipse  
    With ellipse  
        .Width = 150  
        .Height = 75  
        .SetValue(Canvas.LeftProperty, 20.0)  
        .SetValue(Canvas.TopProperty, 30.0)  
        .Fill = New SolidColorBrush(Color.FromArgb(128, 0, 255, 0))  
    End With  
    sender.Children.Add(ellipse)  
End Sub
```



- Jedes XAML-Element korrespondiert mit einem Objekt in der Common Language Runtime (CLR) und kann entsprechend manipuliert werden
- XAML-Attribute werden im „CodeBehind“ als Properties abgebildet
- Spezielle Klassen und Methoden stehen im Anwendungskontext zur Verfügung

→ Event-basierte Aktionen im „CodeBehind“ behandeln (3) [VB.NET]

```
<Canvas>  
  <Button Canvas.Left="30" Canvas.Top="30" Padding="10" Cursor="Hand"  
    Content="GET Request" Click="HoleDaten"/>  
  <TextBlock Canvas.Left="150" Canvas.Top="40" x:Name="ausgabe" Text="..."/>  
</Canvas><!-- Dynamischer Zugriff auf externe Ressourcen, hier PHP-Script -->
```

Button + TextBlock

```
Private Sub HoleDaten(ByVal sender As System.Object, _  
                    ByVal e As System.Windows.RoutedEventArgs)  
  
  Dim wclient As New WebClient  
  
  AddHandler wclient.DownloadStringCompleted, AddressOf DownloadStringCompleted  
  
  Dim address As Uri = New Uri("script.php?x=123", UriKind.Relative)  
  wclient.DownloadStringAsync(address)  
  
End Sub  
  
Private Sub DownloadStringCompleted(ByVal sender As System.Object, _  
                                   ByVal e As DownloadStringCompletedEventArgs)  
  
  If e.Error Is Nothing Then  
    ausgabe.Text = e.Result  
  Else  
    ausgabe.Text = e.Error.Message  
  End If  
  
End Sub
```

Asynchrone
Datenkommunikation
mit dem Server

Ergebnisausgabe im
TextBlock-Element

→ Event-basierte Aktionen im „CodeBehind“ behandeln (4) [VB.NET]

```
<Canvas>
  <ext:Calendar Canvas.Left="30" Canvas.Top="30" x:Name="kalender"
    SelectedDatesChanged="Auswahl"/>
  <TextBlock Canvas.Left="30" Canvas.Top="200" x:Name="datum" Text="..."/>
</Canvas><!-- Abfrage des ausgewählten Kalender-Datums -->
```

Calendar + TextBlock



November 2008						
Mo	Di	Mi	Do	Fr	Sa	So
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Ausgewählt: 05.11.2008

```
Private Sub Auswahl(ByVal sender As System.Object, _
                    ByVal e As System.Windows.Controls.SelectionChangedEventArgs)
    datum.Text = "Ausgewählt: " + kalender.SelectedDate
End Sub
```

Ergebnisausgabe im
TextBlock-Element

Hinweis: Unter „Referenzen“ ist eine Website des Autors mit weiteren Code-Beispielen verlinkt, welche JavaScript- und VB.NET-Ansätze demonstrieren.

Silverlight!

→ Anwendung „Codex Sinaiticus“ (Bibel-Handschriften der Uni-Bibl. Leipzig)

The screenshot displays a digital interface for the 'Codex Sinaiticus' application. On the left, a high-resolution image of a manuscript page is shown, featuring columns of text in a traditional script. The page is labeled 'Ester' at the top. On the right side, a vertical panel contains a German translation of the text, with verse numbers 1.1h through 1.1q. The interface includes a navigation bar at the bottom with icons for home, search, and zoom, and a list of books including Josua, Richter, 1 Chronik, 2 Esra, Ester, Tobit, Judit, 1 Makkabäe, and 4 Makkabäe. A 'Buchauswahl schließen' button is also visible.

<http://www.e-manuscripts.org>

→ Anwendung „Fußball Special zur EM 2008“ von MSN Sport

Fußball Special

EM Startseite | Video-Magazin | News | Fußball Manager | Forum

Turniertag: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

EM Video-Magazin

Bilder des Tages

Statistik Grafiken

3D Tor-Center

GRUPPENSPIELE

Gruppe B

Österreich	0:1	Kroatien
Deutschland	2:0	Polen
Kroatien	2:1	Deutschland
Österreich	1:1	Polen
Polen	0:1	Kroatien
Österreich	0:1	Deutschland

FINALE

Turniertag: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

EM Video-Magazin

Bilder des Tages

Statistik-Grafiken

Deutschland Spanien 0:1

3D Tor-Center

Torschuss-Analyse

Fairness

Zweikampferhalten

Torschussanalyse

<http://euro08magazin.sport.msn.de>

→ Anwendung „FoidosChess“ von Chess Media Services

The screenshot shows the FoidosChess application interface. At the top left, it displays the date and location: "Donnerstag, 23. Oktober 2008, 17:21:09 Bonn, DE". The main background features logos for "EVONIK INDUSTRIES" and "GAZPROM", and the text "WORLD CHESS Championship 2008 German". A purple banner indicates the event dates: "October 14th until November 2nd".

On the right side, there is a navigation menu with icons for "Help", "About", "Messages", and "Exit".

The central focus is a chess game window. It shows two players: "Player1 Player1 (1)" with a timer of "00:59" and "Player2 Player2 (1)" with a timer of "00:43". Below the player information are two video feeds showing the players. A chessboard is displayed below the feeds, with a list of moves on the right: 20 Ba3 Qc3, 21 Rac1 Qxe3, 22 fxe3 f6, 23 Bd6 g5, 24 h3 Kf7, 25 Kf2 Kg6, 26 Ke2 fxe5, 27 dxe5 b6, 28 b4 Rc4. At the bottom of the chessboard window are navigation arrows and a "Shredde" button.

At the bottom of the application, there is a row of eight video thumbnails: "Player1", "Player2", "Ambient", "Roaming Reporter", "German Commentary", "English Commentary" (with the text "This is the english commentator video message"), "Russian Commentary" (with "Adrian Mikhalet OFF AIR"), "Spanish Commentary" (with "Amador Rodrig OFF AIR"), and "Opening Cer".

<http://www.foidoschess.tv/FoidosChess-Demo.aspx>

→ Anwendung „Patient Journey Demonstrator“ von MS Health

The screenshot displays a complex web application interface for patient monitoring. On the left, a 'Chest Pain Clinic' panel lists patients with 'View Results' buttons. A central 'Live Exercise Test Stream' shows 'Stage 4' with a heart rate of 176 and blood pressure of 126/69. Below this is an 'Angiogram Viewer' displaying a 3D heart model and a 2D angiogram of the LAD. A message window from David Hamilton is overlaid on the interface, containing a small ECG snippet and a 'View Live ECG' button. The bottom of the screen features a navigation bar with 'Active Care Pathways (2)', 'Essential Hypertension NICE Guidelines - Step 2', and 'Moderate Depression Stage 4/5 - Maintenance Treatment...'. A search bar for 'Care Pathway Library' is also visible.

[http://www.mscai.net/
PatientJourneyDemonstrator/](http://www.mscai.net/PatientJourneyDemonstrator/)

- Zusammenfassung / Ausblick

- Silverlight 2 ist eine fortgeschrittene und plattformübergreifende RIA-Technologie mit vielfältigen Einsatzmöglichkeiten
- Besonders Entwickler im .NET-Bereich können ihre Erfahrungen und Kenntnisse einbringen und schnell zu Ergebnissen kommen
- Plugin-Verbreitung ist gegenwärtig noch relativ gering
(stärkere Verbreitung ist im Rahmen neuer Silverlight-Anwendungen zu erwarten, denkbar sind auch entsprechende Microsoft-Systemupdates)
- Wünsche für die Zukunft:
 - Bessere Linux-Unterstützung
 - Kamera- und Mikrofon-Anbindung
 - Standalone-Anwendungen in Analogie zu Adobe AIR
 - 3D-Techniken
 - Entwicklung mobiler Anwendungen mit passenden Werkzeugen (→ 2009)
 - Weitere offene Werkzeuge (Eclipse Tools for Silverlight → eclipse4sl.org)
 - Freie Blend-Version („Blend Express“?)

- Referenzen

- Microsoft Silverlight: <http://silverlight.net>
- Feature Matrix (SL1/2): <http://silverlight.net/GetStarted/overview.aspx>
- Werkzeuge: <http://www.microsoft.com/silverlight/resources/tools.aspx>
- XAML: <http://msdn.microsoft.com/en-us/library/ms752059.aspx>
- Moonlight: <http://www.mono-project.com/Moonlight>
- Beispielmaterial: <http://slxlb.c.datenverdrahten.de>

- Kontakt

- E-Mail: thomas.meinike@hs-merseburg.de
- WWW: <http://www.iks.hs-merseburg.de/~meinike/>