



dotnet Cologne 2011

Silverlight 5 Beta 1

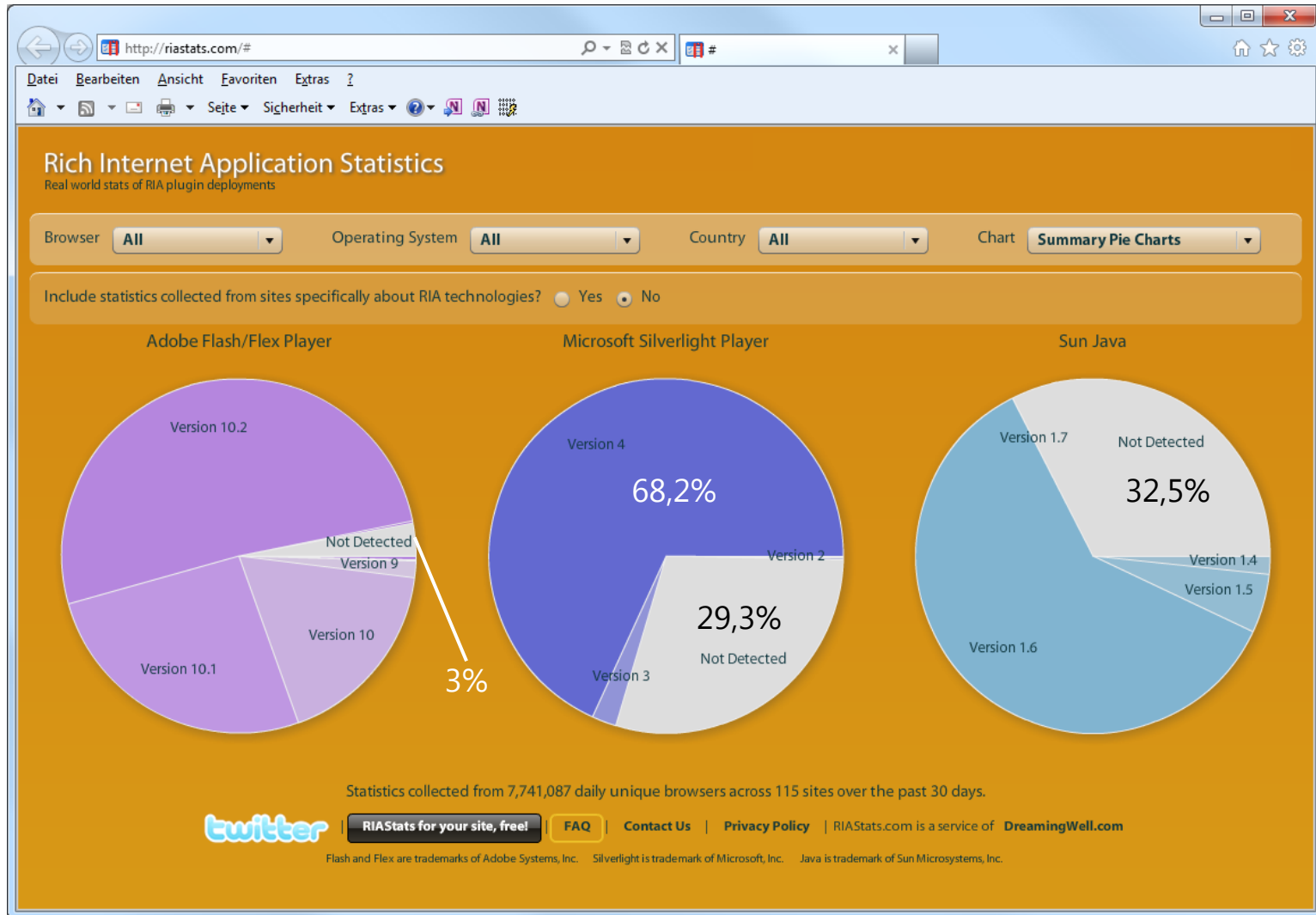
Neues in „Silverlight für den Browser“

Stefan Lange
empira Software GmbH
Stefan.Lange@empira.de
www.st-lange.net

Inhalt

- Voraussetzungen
- Tooling, XAML, Trusted Apps, Demos
- 3D Support
- Ausblick
- Fragen, Diskussion

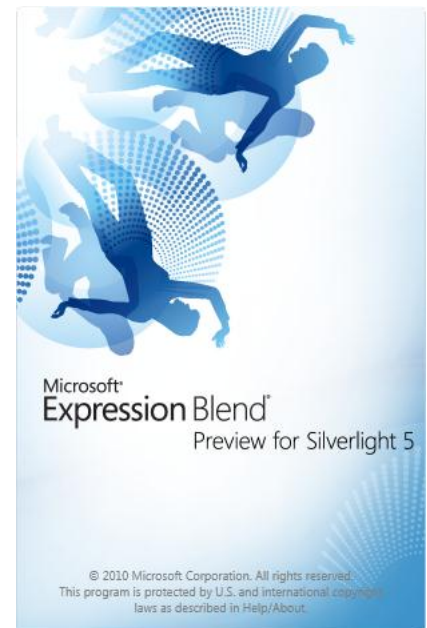
Silverlight heute



Stand: **Mai 2011**

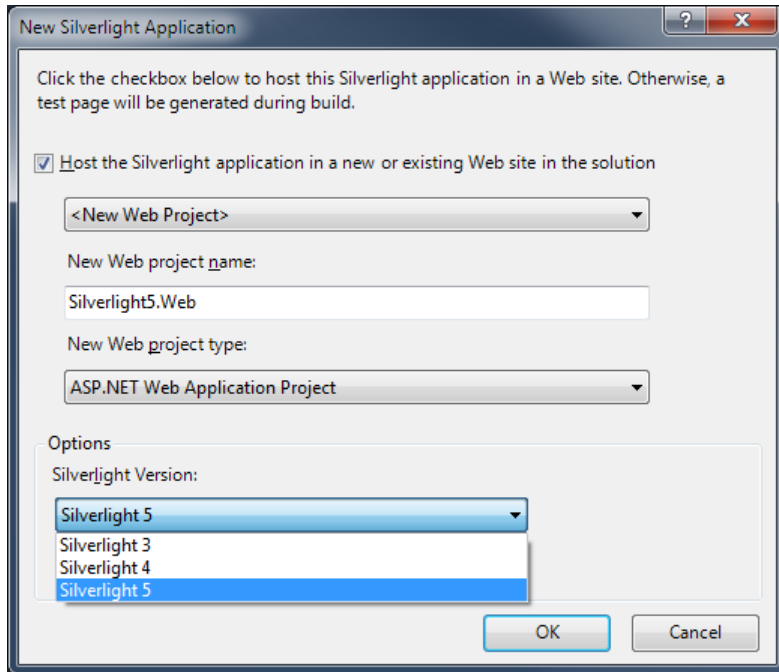
Voraussetzungen

- Visual Studio 2010 mit SP1
- Silverlight 5 Beta 1
<http://www.silverlight.net/getstarted/silverlight-5-beta/>
- Weitere optionale Downloads
 - Silverlight 5 Hilfe
 - Blend Preview für Silverlight 5
 - Control-Toolkit von Silverlight 4 verwenden



Multi Target Unterstützung

Visual Studio 2010 SP1



Beachten:
Generell nur eine
Silverlight Runtime pro
Computer:
Silverlight 5

Quick-Tipp

Silverlight Runtime 4 und 5 gleichzeitig

- Silverlight 4 und Firefox installieren
- Firefox-Verzeichnis mit Silverlight-Plugin und Silverlight Runtime sichern
- Silverlight 5 installieren
- Gesicherte Verzeichnisse zurückkopieren
- Ergebnis: Firefox mit SL 4, IE mit SL 5

*Gehört Silverlight 5 Beta 1 auf
Entwicklungsrechner?*

Ja!

Breaking Changes in Silverlight 5

Neue Features in Silverlight 5

ClickCount

DataContextChanged

Textdarstellung

Policy based Trust

Multiple Windows

Trickplay

WCF RIA Services

Blend für Silverlight 5

Relative Ancestor Binding

Implizite Data Templates

Trusted In-Browser Apps

XNA Framework

Breakpoint in Binding

Uneingeschränkter
Dateizugriff

Binding in Styles

COM In-Browser

Custom Markup Extensions

XNA Sound Effekte

UpdateSource Trigger

64 Bit Support

P/Invoke

ICustomTypeProvider

Verbesserungen im Binding

Platz 2 auf der Silverlight-Wunschliste:

- Binding wie in WPF
(Keine Binding-Hacks mehr)

(<http://dotnet.uservoice.com/forums/4325-silverlight-feature-suggestions>)

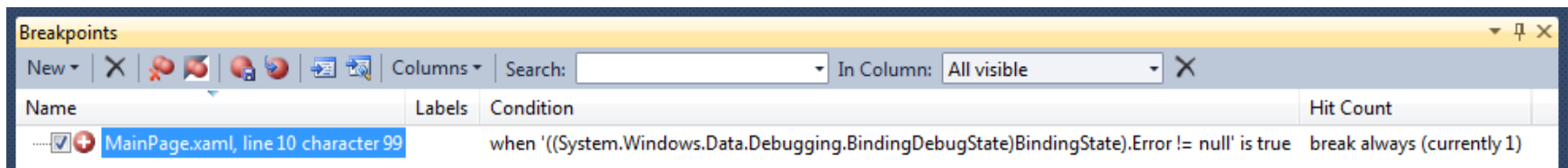
Binding-Breakpoints in XAML

Wie im Code: F9 – Toggle Breakpoint

```
<TextBlock Text="{Binding SomeText}" .../>
```

Binding-Breakpoints in XAML

- Conditional Break (Break nur im Fehlerfall)



```
public class BindingDebugState
{
    public BindingBase Binding { get; internal set; }
    public Expression BindingExpression { get; internal set; }
    public BindingAction Action { get; internal set; }
    public BindingStage LastCompletedStage { get; internal set; }

    public BindingValuePipeline Pipeline { get; internal set; }

    public object[] ValidationErrors { get; internal set; }
    public object Error { get; internal set; }
    public object FinalSource { get; internal set; }
    public override string ToString();
}
```

BindingDebugState

Viele nützliche Infos
zur Laufzeit

Breakpoint in Binding

DEMO

Quick-Tipp

- Breakpoint in Bindings funktionieren auch in Silverlight 4-Projekten 😊

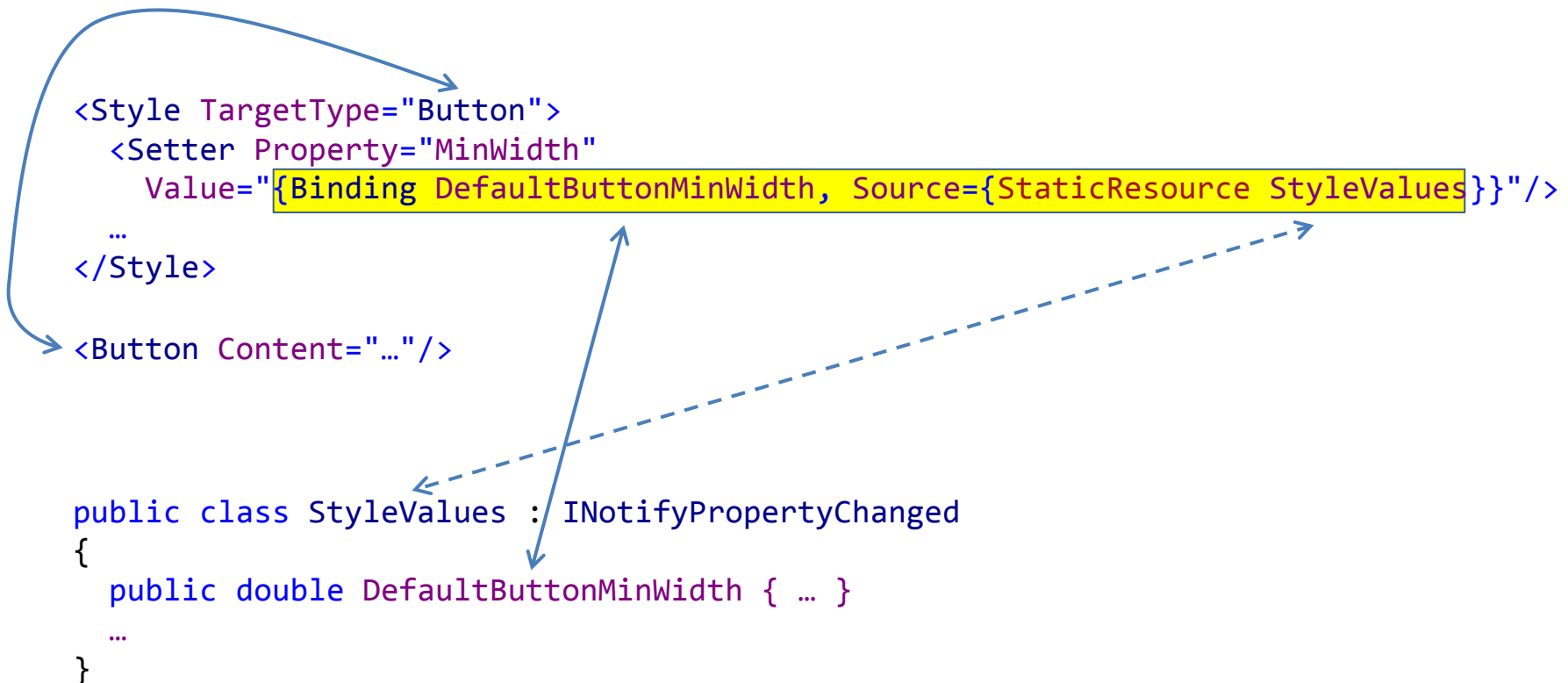
Binding in Styles

- Binding jetzt auch im Setter möglich

```
<Style TargetType="Button">  
  <Setter Property="MinWidth"  
    Value="{Binding DefaultButtonMinWidth, Source={StaticResource StyleValues}}"/>  
  ...  
</Style>
```

```
<Button Content="..."/>
```

```
public class StyleValues : INotifyPropertyChanged  
{  
  public double DefaultButtonMinWidth { ... }  
  ...  
}
```



Binding in Styles

DEMO

Was bringt Binding in Styles?

- Einstellungen lassen sich in Styles auslagern
- Einstellung zur Laufzeit änderbar
- Sehr aufgeräumter XAML Code
- Funktionalität von DynamicResource aus WPF kann damit ersetzt werden

Implicit Data Templates

Hierarchie von Datenklassen über Type-spezifische Templates darstellbar

```
public class DataItem
{
    public string SomeText { get; set; }
}

public class RedDataItem : DataItem
{
    ...
}

public class BlueDataItem : DataItem
{
    ...
}
```

Silverlight 4

Nur ein Template pro
Control möglich

Implicit Data Templates

- Für jeden Datentyp ein Template

```
<ListBox.Resources>
```

```
  <DataTemplate DataType="local:DataItem">  
    <TextBlock Text="{Binding SomeText}"/>  
  </DataTemplate>
```

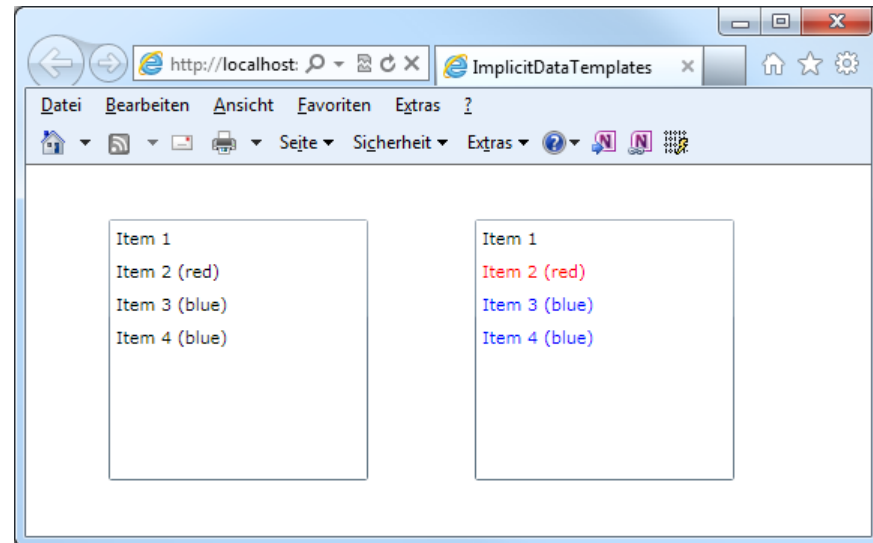
```
  <DataTemplate DataType="local:RedDataItem">  
    <TextBlock Text="{Binding SomeText}" Foreground="Red"/>  
  </DataTemplate>
```

```
  <DataTemplate DataType="local:BlueDataItem">  
    <TextBlock Text="{Binding SomeText}" Foreground="Blue"/>  
  </DataTemplate>
```

```
</ListBox.Resources>
```

Implicit Data Templates

DEMO



Was ist mit Template Selector?

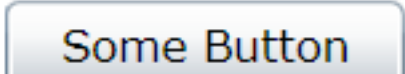
- Auswahl des Templates per Callback
- Nicht in Silverlight 5
- Lässt sich aber auch per Code lösen

Relative Source Binding

- Nützliches WPF-Feature für Bindings in Templates

```
<Grid x:Name="LayoutRoot" Background="White" Margin="40">
  <ContentControl Tag="SomeValue" > ← 2
    <Controls:HeaderedContentControl > ← 1
      <Controls:HeaderedContentControl.Header >
        <TextBlock Text="{Binding Tag,
          RelativeSource={RelativeSource AncestorType=ContentControl,
            AncestorLevel=2}}" />
      </Controls:HeaderedContentControl.Header >
      <Button Width="100">Some Button</Button>
    </Controls:HeaderedContentControl >
  </ContentControl >
</Grid >
```

SomeValue

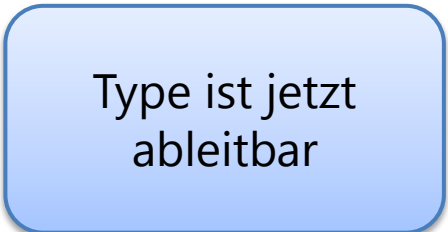


ICustomTypeProvider

Binden an laufzeitgenerierte „Properties“

```
public class MyDataClass : ICustomTypeProvider
{
    public Type GetCustomType()
    {
        ...
    }

    public void AddProperty(string name, Type type)
    { ... }
}
```



Type ist jetzt ableitbar

- Spart Unmengen trivialer Properties
- Kein Binding an Type dynamic möglich
- Ähnlich ICustomPropertyDescriptor aus WPF

Quick-Tipp

Zu `ICustomDataProvider` siehe hier:

http://blogs.msdn.com/b/silverlight_sdk/archive/2011/04/25/binding-to-dynamic-properties-with-icustomtypeprovider-silverlight-5-beta.aspx

Custom Markup Extensions

- Bisher nur fest eingebaute Markup Extensions
 - {Binding... }
 - {StaticResource... }
- Bisher „hart codiert“ im XAML-Parser

Custom Markup Extensions

Ableiten von MarkupExtension

```
public class SampleExtension : MarkupExtension
{
    public override object ProvideValue(IServiceProvider serviceProvider)
    {
        ...
        return "1234";
    }

    public string SomeValue { get; set; }
}
```

```
<TextBlock Text="{local:Sample SomeValue='abc'}" ... />
```

Zugriff auf
RootObject
TargetObject
TargetProperty

ProvideValue wird genau einmal vom XAML-Parser aufgerufen

Custom Markup Extensions

Vielfältige Einsatzmöglichkeiten

- Zugriff auf globale Datenstrukturen
- Zugriff auf resx-Daten zur Lokalisierung
- x:Static Extension aus WPF nachbauen
`Text="{local:Static Member=local:StaticClass.AStaticValue}"`
- Formelparser
- Eigenes Binding, z.B. an Felder

Custom Markup Extensions

DEMO

Fazit Bindings

- Großer Schritt Richtung WPF
- Wichtigste Features alle vorhanden
- Silverlight 5 zielt damit auf komplexe Business-Anwendungen

Neues Trusting-Level

Historie

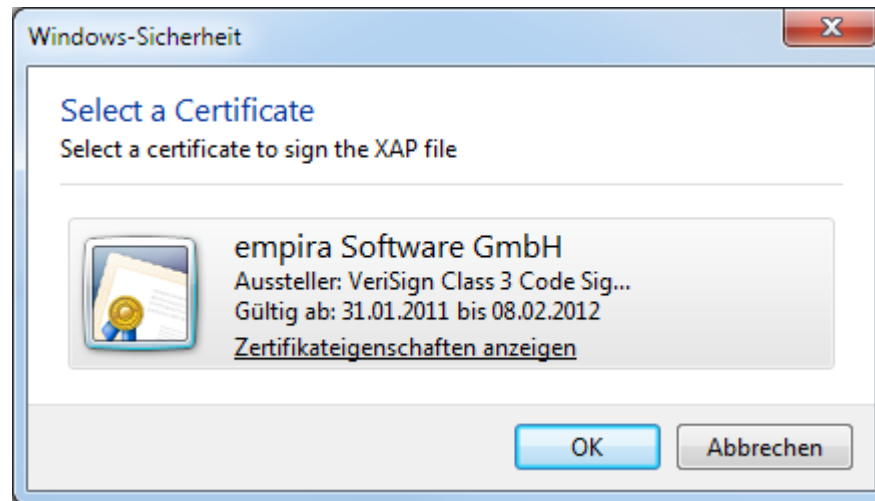
- Bis Silverlight 3: Browser-basierter Sandbox
- Silverlight 4: Out-of-Browser und Trusted Apps

Neu

- Silverlight 5: In-Browser Trusted Apps

Software Publisher Certificate

Zertifikate sind wichtig in Silverlight 5



Quick-Tipp

- Praktisch während der Entwicklung:
„localhost“ wird immer vertraut

- Notwendiger Hack für die Beta 1:
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Silverlight\
AllowElevatedTrustAppsInBrowser=1`

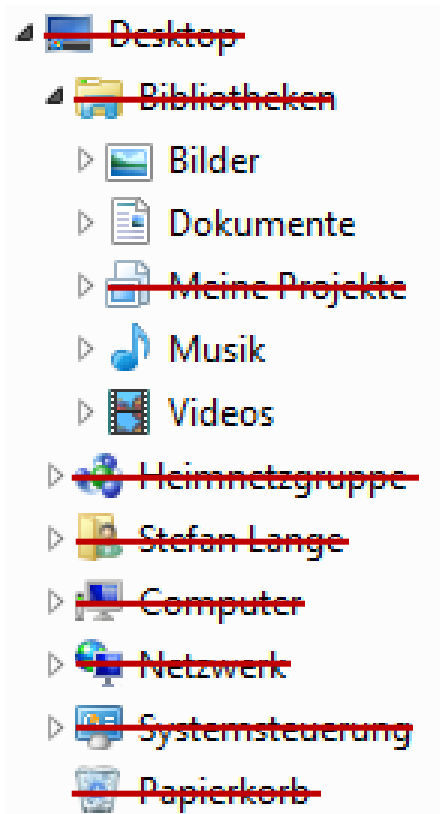
Quick-Tipp

- Certificat von VeriSign kaufen
 - Der Name ist sowohl bei Kunden als auch Admins sehr bekannt
 - Von Microsoft als einziges anerkannt für Zertifizierungen
 - Keine Probleme mit Firewalls etc.
- Kauftipp zu VeriSign
 - Nicht einfach online kaufen
 - Anrufen und penetrant nachfragen
 - Oft gibt es ein günstigeres „special Offer“ 😊

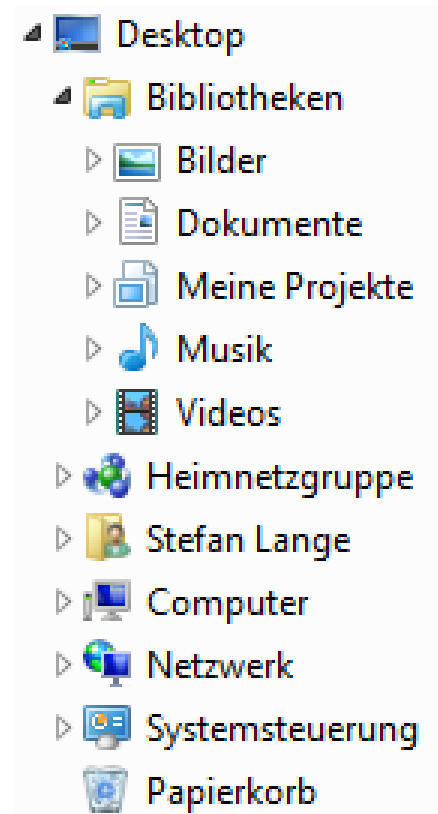
Dateizugriffsbeschränkung

Vergleich Trusted Apps

Silverlight 4



Silverlight 5



Vergleich der Trust-Level

Mehr Freiheiten in Silverlight 5

| | Silverlight Sandbox | Silverlight 4 Trusted | Silverlight 5 Trusted |
|--------------------------------|---------------------|-----------------------|-----------------------|
| Prompt-free | nein | ja | ja |
| Cross Domain Zugriff | nein | ja | ja |
| Tastaturzugriff bei Vollbild | nein | ja | ja |
| COM | nein | ja | ja |
| Uneingeschränkter Dateizugriff | nein | teilweise | ja |
| P/Invoke | nein | nein | ja |
| Trusted „In-Browser“ | nein | nein | ja |

Trust Policy

- Policy-basiertes Modell in Entwicklung
- Admin konfiguriert Policy (Authenticode)
- Aktueller Stand in Beta 1:
 - trusted
 - untrusted

WebBrowser Control

- Eingeführt mit Silverlight 4
- Jetzt auch In-Browser lauffähig
- Ergänzen von Silverlight durch HTML

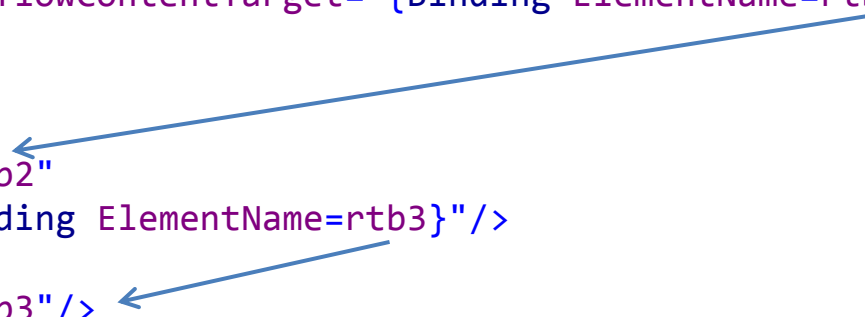
In-Browser WebBrowser Control

DEMO

RichTextBoxOverflow

- Verkettung der Inhalte von TextControls

```
<RichTextBox x:Name="rtb1" OverflowContentTarget="{Binding ElementName=rtb2}">  
  <Paragraph>...</Paragraph>  
</RichTextBox>
```



The diagram illustrates the chain of `OverflowContentTarget` bindings between three `RichTextBox` controls:

- `rtb1` (top) has `OverflowContentTarget="{Binding ElementName=rtb2}"`. A blue arrow points from this attribute to the `rtb2` element below.
- `rtb2` (middle) has `OverflowContentTarget="{Binding ElementName=rtb3}"`. A blue arrow points from this attribute to the `rtb3` element below.
- `rtb3` (bottom) has no `OverflowContentTarget` attribute.

```
<RichTextBoxOverflow x:Name="rtb2"  
  OverflowContentTarget="{Binding ElementName=rtb3}"/>
```

```
<RichTextBoxOverflow x:Name="rtb3"/>
```

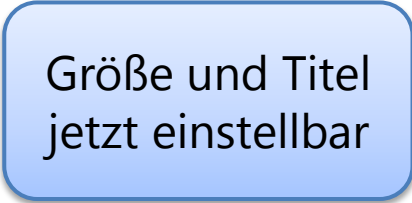
RichTextBoxOverflow

DEMO

Multi Window Support

- Window ist jetzt instanziiierbar

```
var window = new Window { WindowState = WindowState.Normal,  
  
Visibility = Visibility.Visible };  
window.Content = new MainPage();  
window.Width = 400;  
window.Height = 300;  
window.Title = "New Window";
```



Größe und Titel
jetzt einstellbar

Natürlich nur Out-Of-Browser verwendbar

Multi Window Support

DEMO

Der Damm ist gebrochen

POST BETA 1

P/Invoke ruft Nativecode auf

Wie in .NET üblich

```
[DllImport("kernel32.dll")]  
static extern bool TerminateProcess(IntPtr hProcess, uint uExitCode);
```

*Damit wird auch das Laden von eigenen
Native-Code DLLs oder .NET möglich*

P/Invoke

- Einfacher zu verwenden als COM
- Nur für Trusted Apps
- Nur für Windows (nicht Mac 😞)

Silverlight x64

POST BETA 1

- Silverlight 5 auch als 64-Bit-Version
- Vorteile von 64-Bit:
 - Zugriff auf 64-Bin COM-Objekte
 - Läuft in 64-Bit-Browsern
 - Sidebar auf 64-Bit-Windows
 - Zugriff auf viel Speicher

Verbesserung beim Druck

POST BETA 1

- Vektorbasiertes Drucken
 - Auf Basis von PostScript (nicht etwa XPS)

Pivot-Viewer

- Pivot-Viewer Teil des SDK
 - Besser konfigurier- und erweiterbar

Pivot-Viewer aus Silverlight 4

DEMO

Echtes 3D

- Platz 3 der Feature-Wunschliste
- Basiert auf XNA (nicht WPF)
- Microsoft.Xna Namespace



Voraussetzungen

- Silverlight 5 Beta 1
- Optional
 - DirectX SDK (Juni 2010)

Solar Wind

DEMO

DrawingSurface

Das Control für die Ausgabe der Gameloop

```
<Grid x:Name="LayoutRoot" Background="Green">  
  <DrawingSurface Draw="OnDraw" SizeChanged="OnSizeChanged" />  
</Grid>
```

Integration in Silverlight UI

```
void OnDraw(object sender, DrawEventArgs args)
{
    // Draw 3D scene.
    scene.Draw(args.GraphicsDevice, args.TotalTime);

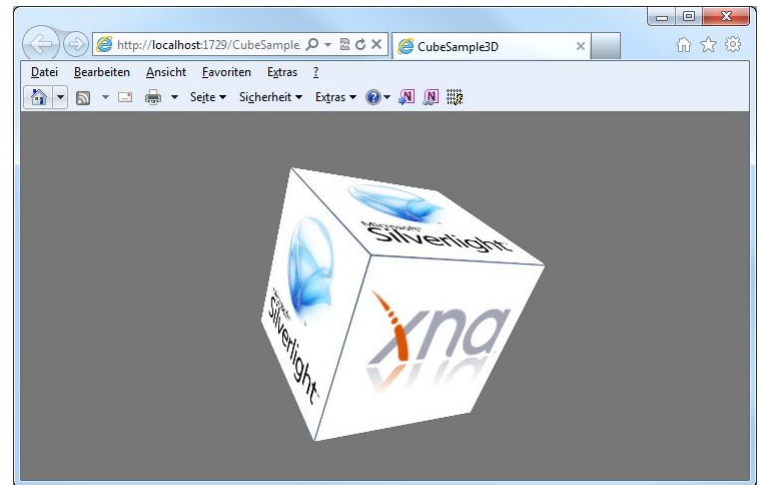
    // Invalidate to get a callback next frame.
    args.InvalidateSurface();
}

// Update the aspect ratio of the scene based on the
// dimensions of the surface.
private void OnSizeChanged(object sender, SizeChangedEventArgs e)
{
    DrawingSurface surface = sender as DrawingSurface;
    scene.AspectRatio = (float)surface.ActualWidth / (float)surface.Actual
Height;
}
```

Läuft in
eigenem XNA-
Thread

Cube Sample 3D

DEMO

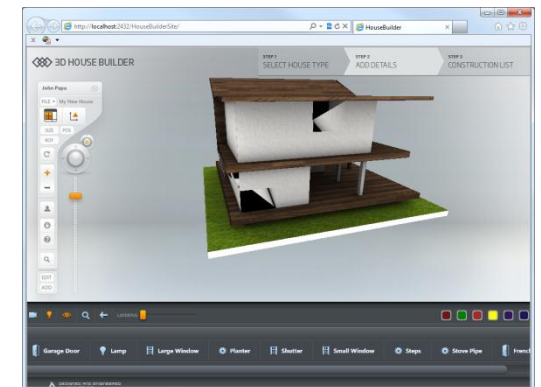
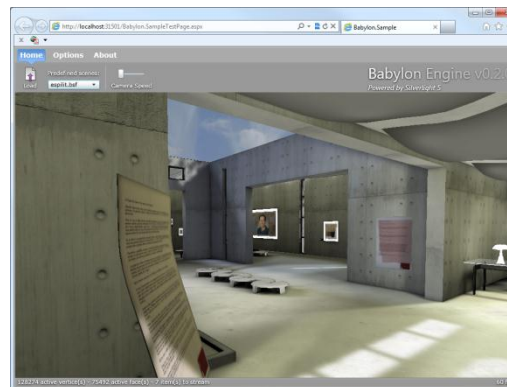


3D Beispielanwendungen

- <http://aarononeal.info/?p=125>

Babylon Engine, 3D HouseBuilder

DEMO



Erscheinungstermin

- Zweite Jahreshälfte
- Wahrscheinlich vorher Beta 2

Fazit/Ausblick

- Weniger cross Plattform, mehr Windows
- Immer näher an WPF-Funktionalität
- Focus auf „High End“ RIA-Anwendungen (PivotViewer, LightSwitch, ...)
- Vermutlich wichtige Basis in Windows 8

Fragen?

Hat Euch der Vortrag gefallen?
Ich freue mich auf Euer Feedback!

Vielen Dank!

Stefan Lange