BASTA!

.NET, WINDOWS, VISUAL STUDIO

Thomas Claudius Huber

Alles, was Sie über XAML wissen müssen

Thomas Claudius Huber

- Principal Consultant @ Trivadis AG
  - Trainer, Coach, Developer, Architect
  - www.thomasclaudiushuber.com
- Spezialisiert auf WPF, XAML, WinApps
- Autor der umfassenden Handbücher zu WPF, Silverlight und Windows Store Apps

Microsoft® MVP Most Valuable Professional

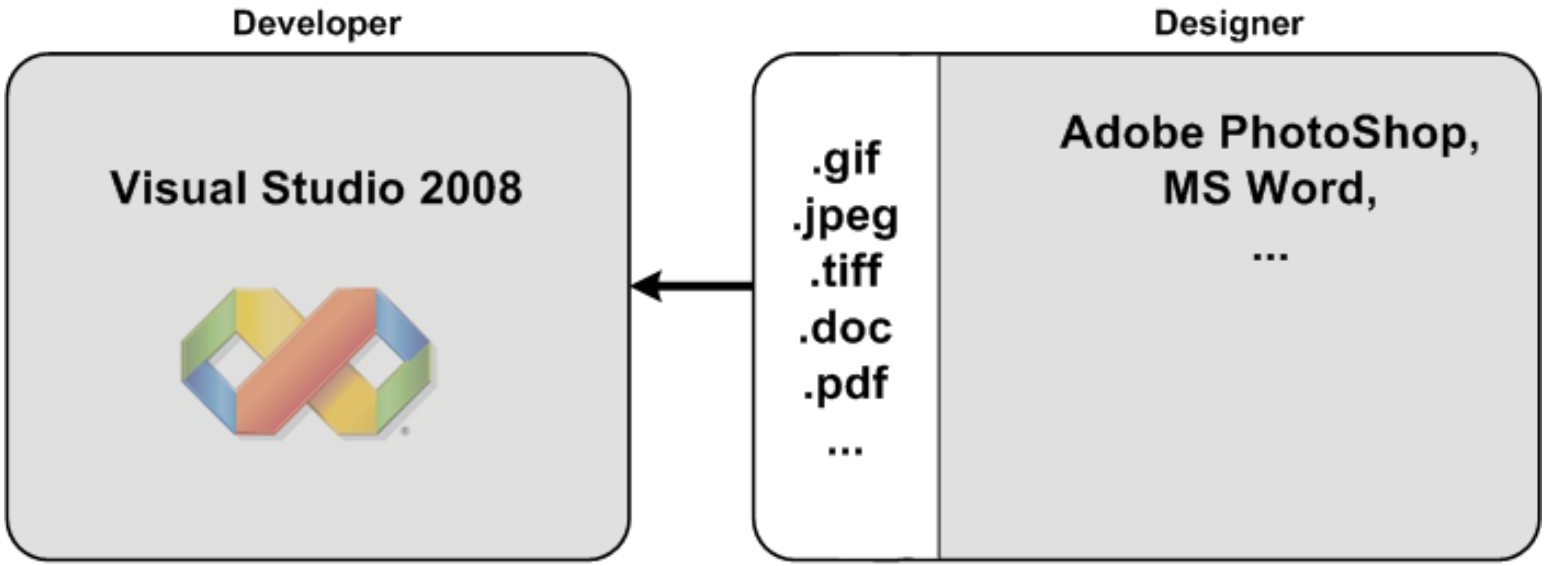
Sessioninhalt

- XAML Basics
- Elemente und Attribute
- Magie
  - TypeConverter
  - Markup Extensions
  - …

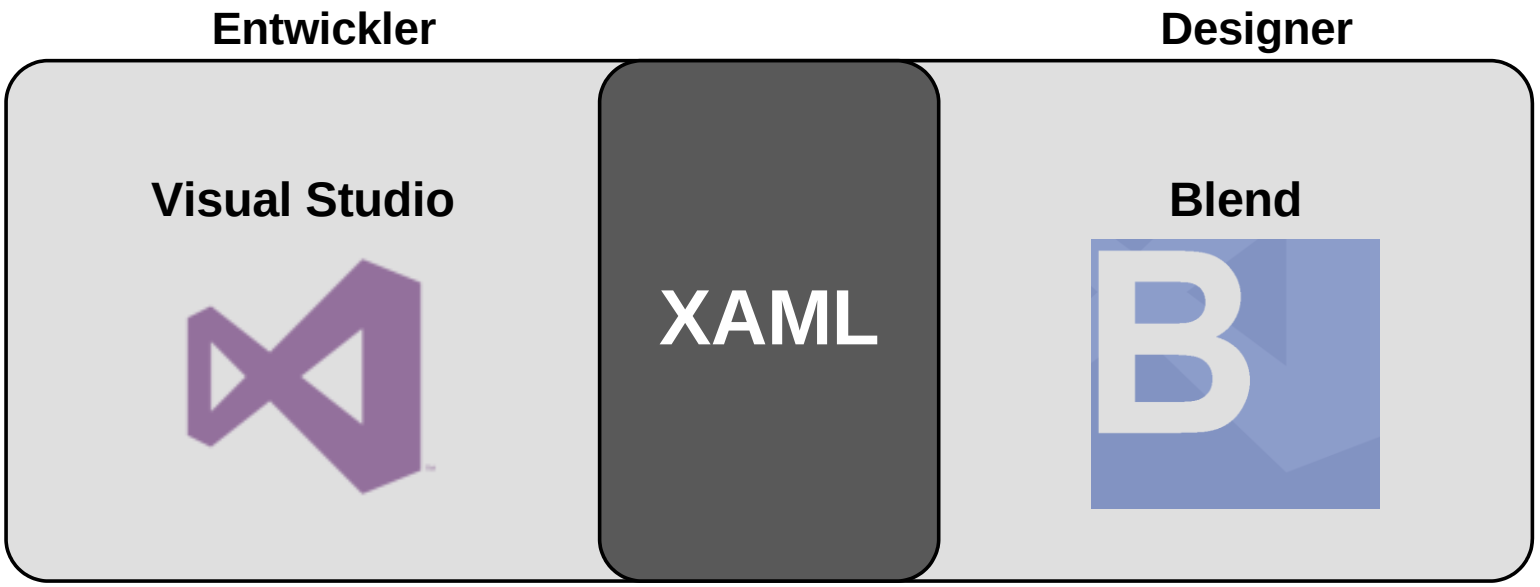
XAML

- Steht für eXtensible Application Markup Language
- Eine XML-basierte Markup Sprache zum Instantiieren von Objekten:
  - Wurde sogar als Serialisierungsformat erstellt
- WPF nutzt XAML als UI-Beschreibungssprache

Warum XAML?


Developer
Designer
Visual Studio 2008
.gif
.jpeg
.tiff
.doc
.pdf
...
Adobe PhotoShop,
MS Word,
...

Workflow mit XAML



Sessioninhalt

- XAML Basics
- Elemente und Attribute
- Magie
  - TypeConverter
  - Markup Extensions
  - …

Properties setzen

- Attribut-Syntax

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

- Property-Element-Syntax

```
<Button>
 <Button.Content>
  OK
 </Button.Content>
</Button>
```

- Implicit as content of element

```
<Button>
 OK
</Button>
```

Demo

Properties setzen

- Implizit als Inhalt des Elements

```
<Button>
 OK
</Button>
```

- Suche via ContentPropertyAttribute

```
[ContentPropertyAttribute(„Content")]
public class ContentControl:Control{...}
```

Demo

- Attached-Property-Syntax

```
<Canvas Width="300" Height="200">
  <TextBox Canvas.Left="50" Canvas.Top="20" Text="Welcome"/>
</Canvas>
```

Klassen-Lookup

- Elemente werden über XML-Namespace zugeordnet

```
<Window x:Class="WpfApplication4.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" .../>
```

- XML-Namespace wird auf Assembly-Ebene mehreren CLR-Namespaces zugeordnet
- Zuordnung findet via XmlnsDefinitionAttribute statt

Demo

1:1-Namespace-Mapping

- Xmlns-Attribut mit Assembly

```
xmlns:alias ="clr-namespace:yourCLRNamespace;assembly=yourAssembly"
```

- CLR-Namespace und XAML in derselben Assembly:

```
xmlns:alias ="clr-namespace:yourCLRNamespace"
```

- Objekte mit gewähltem Alias instantiieren

```
<alias:YourClass/>
```

1:n-Namespace-Mapping

- XmlnsDefinitionAttribute auf Assembly nutzen
- XmlnsPrefix zum Vorschlagen eines Alias
- Ideal für Bibliotheken mit mehreren CLR-Namespaces

Xmlns von XAML

- x-Alias

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

- Ist dem CLR-Namespace System.Windows.Markup zugeordnet
- Enthält Compiler-Direktiven
  - x:Class
  - k:Key
  - x:Code
  - x:Name

Sessioninhalt

- XAML Basics
- Elemente und Attribute
- Magie
  - TypeConverter
  - Markup Extensions
  - …

TypeConverter

- Margin-Property in XAML setzen

```
<Button Margin="10" Content="Book TVD-course"/>
```

- Margin-Property in C#

```
Button btn = new Button();
btn.Margin = new Thickness(10);
btn.Content = "Book TVD-course";
```

- XAML wandelt den String «10» in ein Thickness-Objekt um

TypeConverter

- Thickness-Struct hat TypeConverterAttribute

```
[TypeConverterAttribute(typeof(ThicknessConverter))]
public struct Thickness:IEquatable<Thickness>{...}
```

- XAML-Parser sucht nach diesem Attribut
  - ThicknessConverter wird instanstiiert und konvertiert String zu Thickness-Objekt
- Eigene TypeConverter von TypeConverter ableiten

Demo

Markup Extensions

- Klassem zum Erweitern von XAML:
  - Binding, StaticResource, …
- Attribut-Syntax: mit geschweiften Klammern

```
<Slider x:Name="sli"/>
<Button Content="{Binding Path=Value, ElementName=sli}"/>
```

- Alternativ zur Attribut-Syntax auch als Objektelement

```
<Slider x:Name="sli"/>
<Button>
 <Binding Path="Value" ElementName="sli"/>
</Button>
```

Demo

XAML & Collections

- IList und IDictionary werden unterstützt
- Bei IList wird die Add-Methode aufgerufen
- Bei IDictionary wird das x:Key-Attribut genutzt
  - XAML

```
<ResourceDictionary>
 <SolidColorBrush x:Key="redBrush" Color="Red"/>
</ResourceDictionary>
```

  - C#

```
ResourceDictionary dictionary = new ResourceDictionary();
SolidColorBrush brush = new SolidColorBrush();
brush.Color = Colors.Red;
dictionary.Add("redBrush", brush);
```

Demo

XAML dynamisch laden

- XAML ist ein Serialisierungsformat
- Lesen via XamlReader
- Schreiben via XamlWriter

Demo

Sessioninhalt

- XAML Basics
- Elemente und Attribute
- Magie
  - TypeConverter
  - Markup Extensions
  - …

Danke

Twitter: @thomasclaudiush

Homepage: www.thomasclaudiushuber.com

Mail: thomas.huber@trivadis.com

Slides/Demos:
www.thomasclaudiushuber.com/blog