

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

XAML Deep Dive

Mehr als "nur" WPF

Mathias Raacke

Nevelop GmbH

Mehr als nur WPF... XAML Deep Dive

- Anwendungsgebiete von XAML
 - Welche Einsatzgebiete gibt es neben WPF / Silverlight?
- XAML Services in .NET 4
 - Wie kann man mit XAML Dateien arbeiten?
- Voraussetzungen
 - Erfahrung mit XAML (WPF- , Silverlightentwicklung)

Was ist XAML?

Extensible Application Markup Language

XAML ist ein Offener Standard

- XAML 2006
- WPF Schema
- Silverlight Schema
- XAML 2009
- WPF Schema 2010
- Silverlight Schema 2010

Basiert auf XML

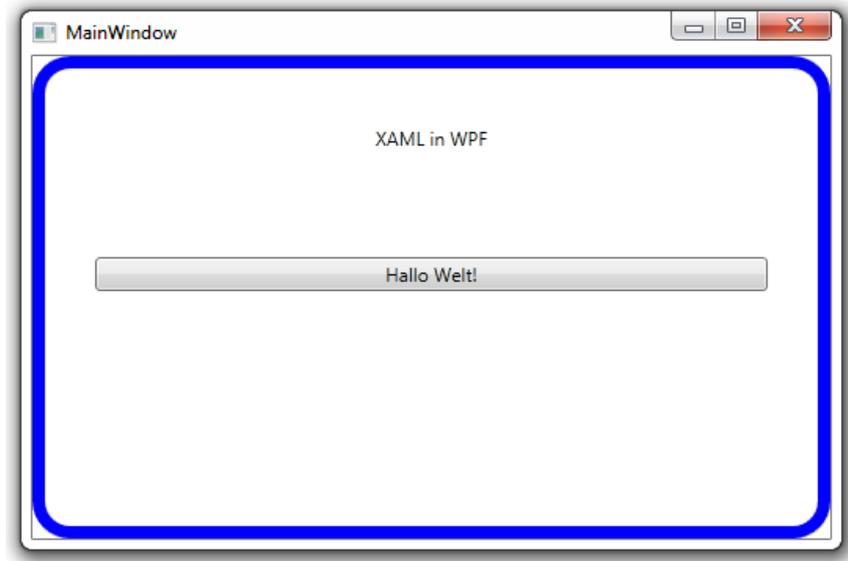
- Ist aber spezialisierter
- Muss nicht zwangsläufig XML verwenden
- Beschreibt Objektgraphen

Wo wird XAML eingesetzt?

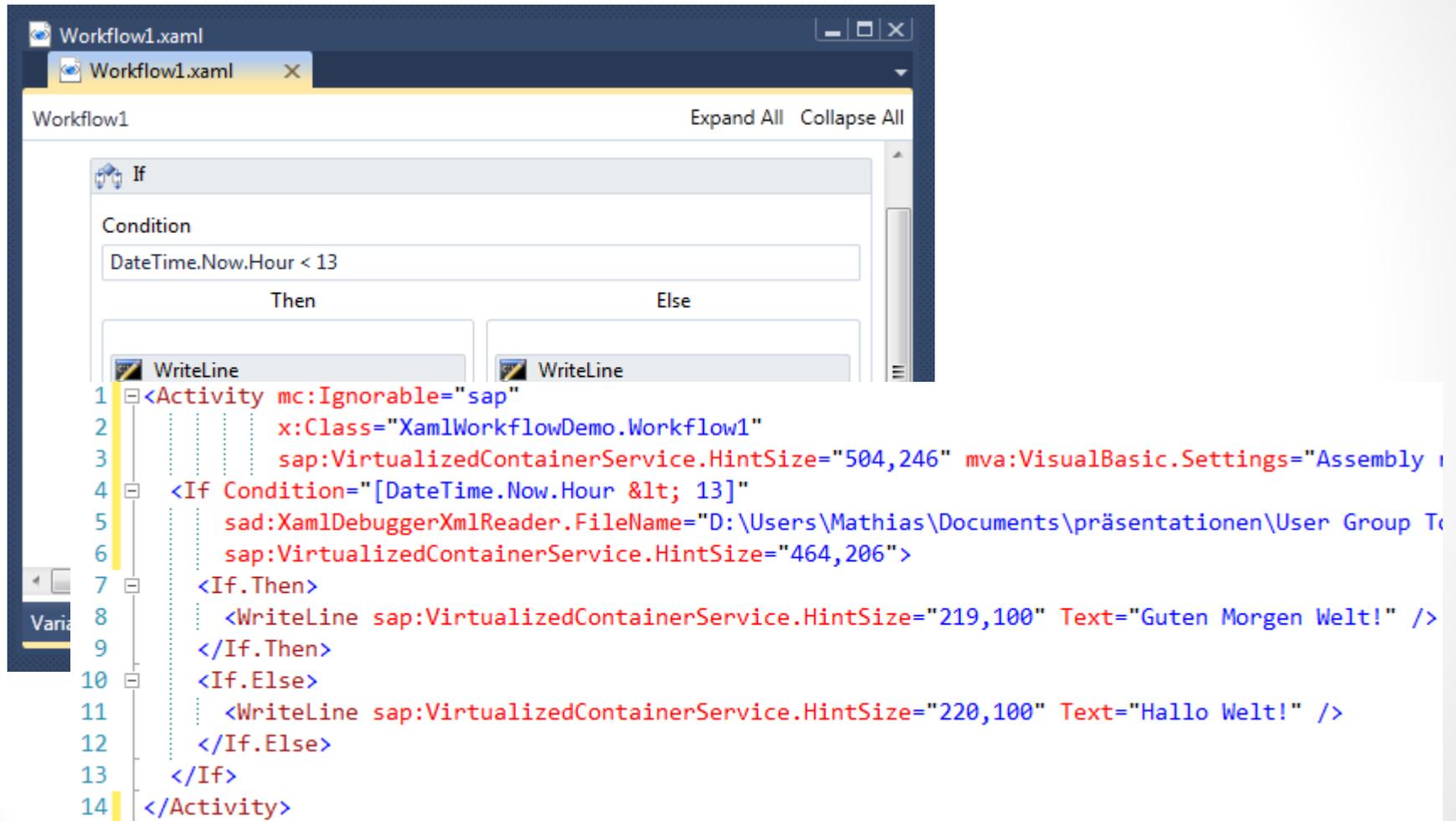
- Windows Presentation Foundation
- Silverlight
- Windows Phone
-?

XAML in WPF

```
1 <Window x:Class="WpfXamlDemo.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4       Title="MainWindow" Height="350" Width="525">
5   <Border CornerRadius="20"
6         BorderBrush="Blue"
7         BorderThickness="8">
8       <StackPanel>
9         <Label HorizontalAlignment="Center" Margin="32">
10            XAML in WPF
11        </Label>
12        <Button Content="Hallo Welt!" Margin="32" />
13      </StackPanel>
14    </Border>
15 </Window>
```



XAML in der Workflow Foundation

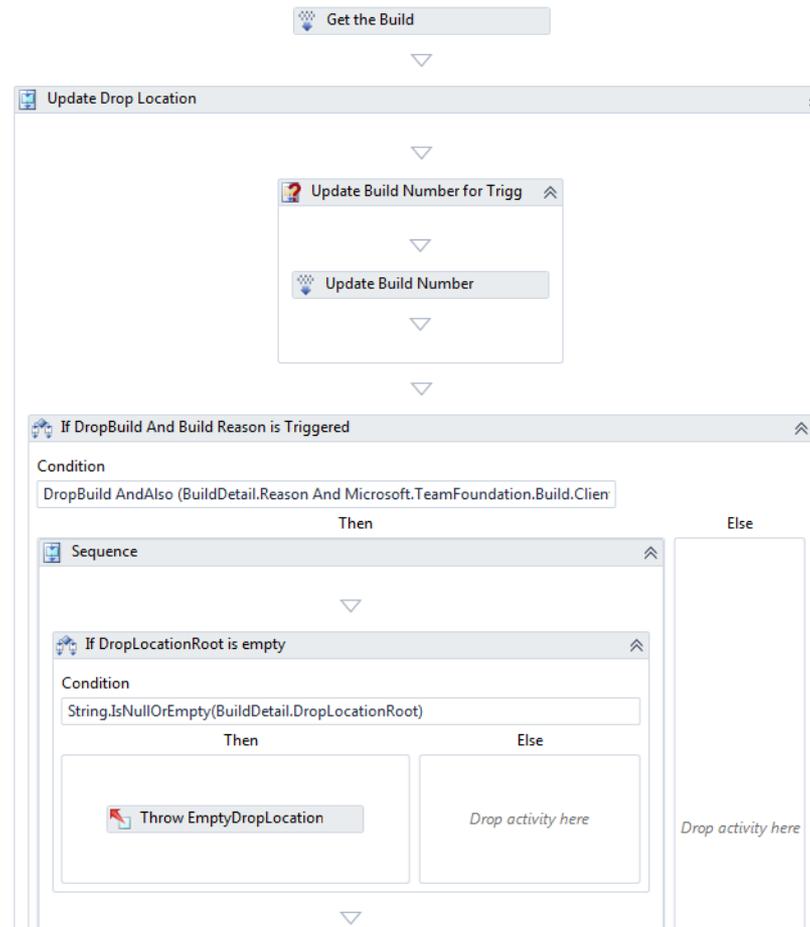


The screenshot displays a Visual Studio window titled 'Workflow1.xaml'. The main area shows a workflow diagram with an 'If' activity. The 'Condition' property is set to 'DateTime.Now.Hour < 13'. The 'Then' branch contains a 'WriteLine' activity with the text 'Guten Morgen Welt!'. The 'Else' branch contains a 'WriteLine' activity with the text 'Hallo Welt!'. Below the diagram, the XAML code is shown, which defines the workflow structure using XML tags.

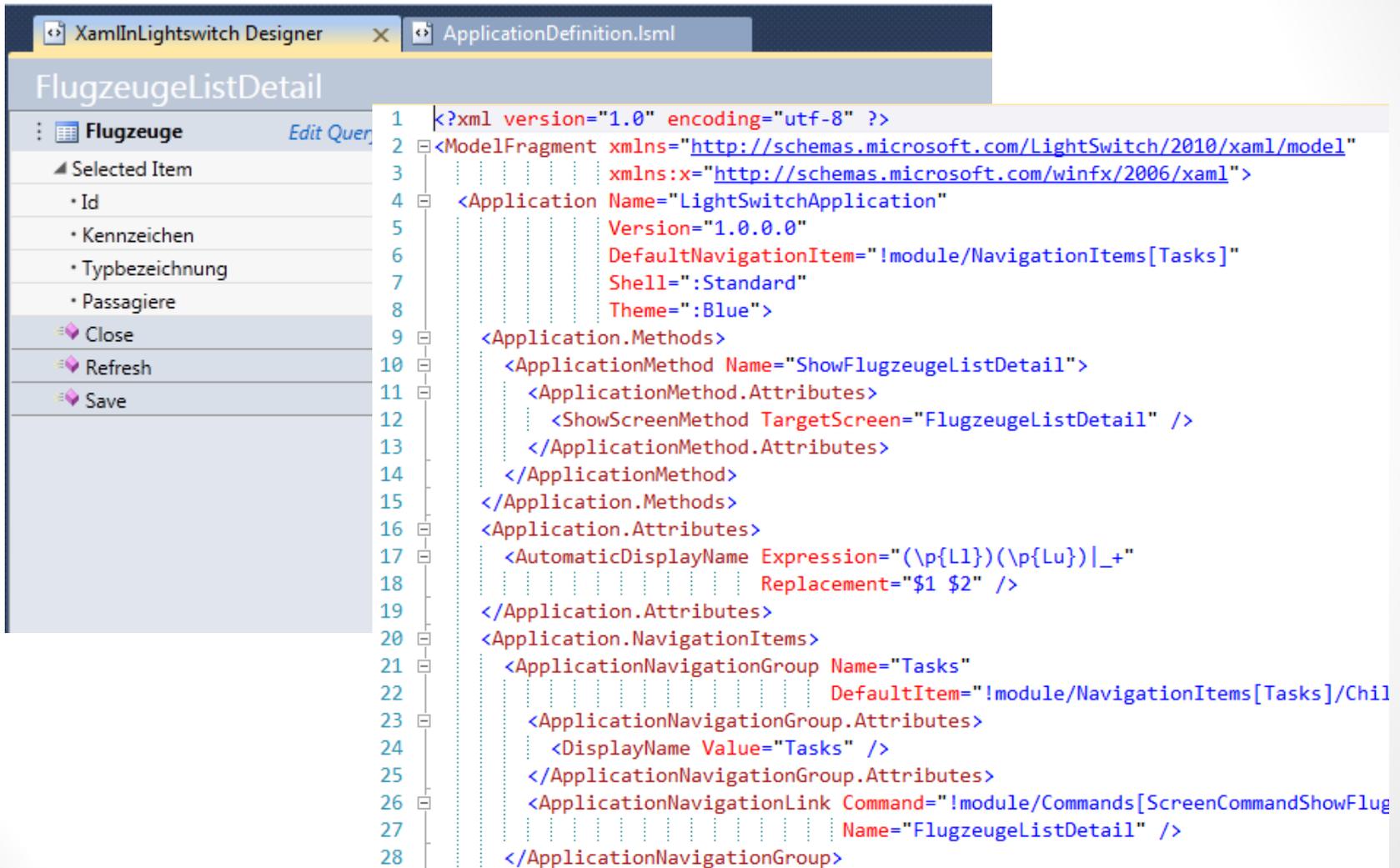
```
1 <Activity mc:Ignorable="sap"
2     x:Class="XamlWorkflowDemo.Workflow1"
3     sap:VirtualizedContainerService.HintSize="504,246" mva:VisualBasic.Settings="Assembly
4 <If Condition="[DateTime.Now.Hour &lt; 13]"
5     sad:XamlDebuggerXmlReader.FileName="D:\Users\Mathias\Documents\präsentationen\User Group To
6     sap:VirtualizedContainerService.HintSize="464,206">
7 <If.Then>
8     <WriteLine sap:VirtualizedContainerService.HintSize="219,100" Text="Guten Morgen Welt!" />
9 </If.Then>
10 <If.Else>
11     <WriteLine sap:VirtualizedContainerService.HintSize="220,100" Text="Hallo Welt!" />
12 </If.Else>
13 </If>
14 </Activity>
```

XAML im Team Build

Team Build basiert auf der Workflow Foundation



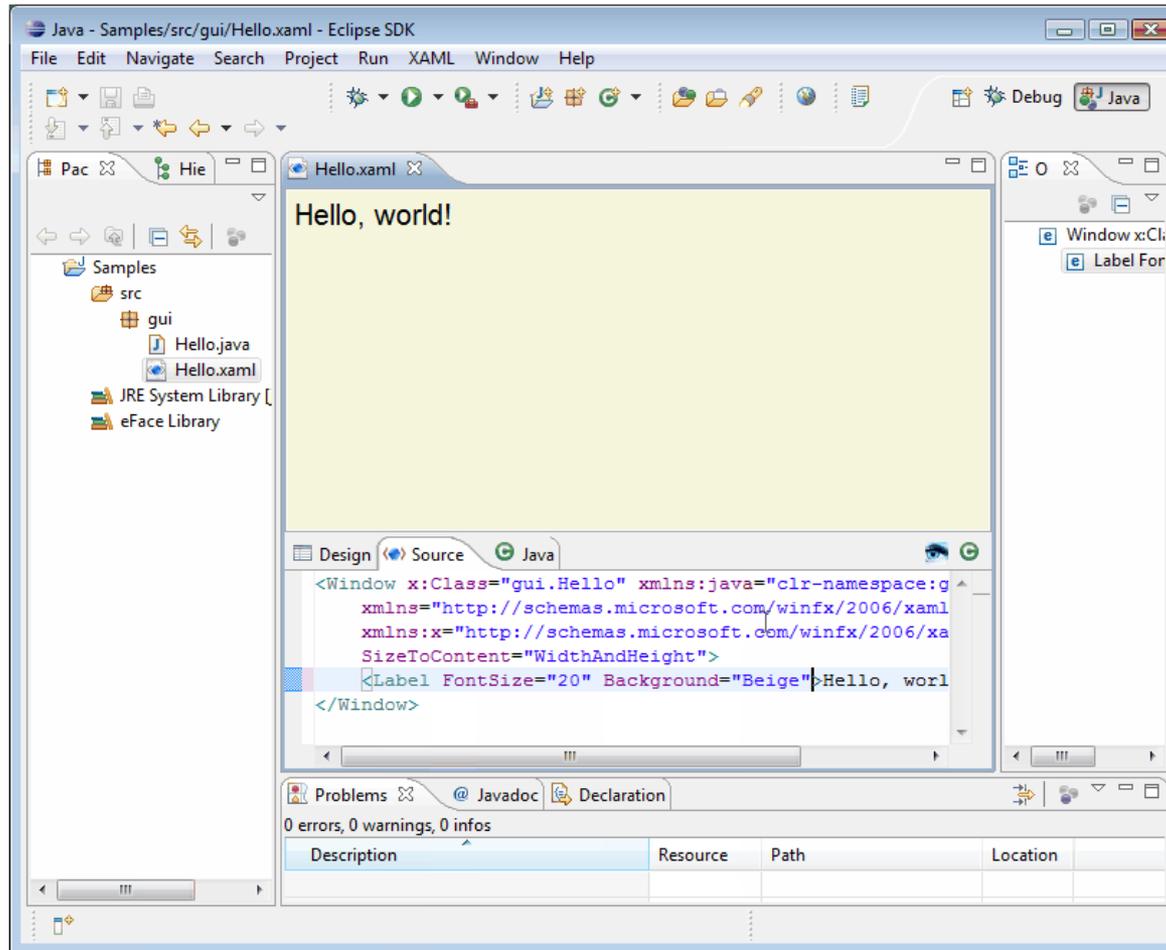
XAML in LightSwitch



The screenshot displays the XamlInLightswitch Designer interface. On the left, a tree view shows the 'Flugzeuge' screen with a list of items: Selected Item, Id, Kennzeichen, Typbezeichnung, Passagiere, Close, Refresh, and Save. The main area shows the 'ApplicationDefinition.Isml' code, which defines the application structure and navigation.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ModelFragment xmlns="http://schemas.microsoft.com/LightSwitch/2010/xaml/model"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
4   <Application Name="LightSwitchApplication"
5     Version="1.0.0.0"
6     DefaultNavigationItem="!module/NavigationItems[Tasks]"
7     Shell=":Standard"
8     Theme=":Blue">
9     <Application.Methods>
10      <ApplicationMethod Name="ShowFlugzeugeListDetail">
11        <ApplicationMethod.Attributes>
12          <ShowScreenMethod TargetScreen="FlugzeugeListDetail" />
13        </ApplicationMethod.Attributes>
14      </ApplicationMethod>
15    </Application.Methods>
16    <Application.Attributes>
17      <AutomaticDisplayName Expression="(\p{L1})(\p{Lu})|_+"
18        Replacement="$1 $2" />
19    </Application.Attributes>
20    <Application.NavigationItems>
21      <ApplicationNavigationGroup Name="Tasks"
22        DefaultItem="!module/NavigationItems[Tasks]/Child" />
23      <ApplicationNavigationGroup.Attributes>
24        <DisplayName Value="Tasks" />
25      </ApplicationNavigationGroup.Attributes>
26      <ApplicationNavigationLink Command="!module/Commands[ScreenCommandShowFlugzeugeListDetail]"
27        Name="FlugzeugeListDetail" />
28    </ApplicationNavigationGroup>
```

XAML in Java / Eclipse



Weitere Anwendungsfälle

- Visual Studio 2010
 - Fenster Layout Einstellungen
 - C++ Build Properties
- Verschiedene Anwendungen
 - Speichern von Konfigurationsinformationen

XAML Services

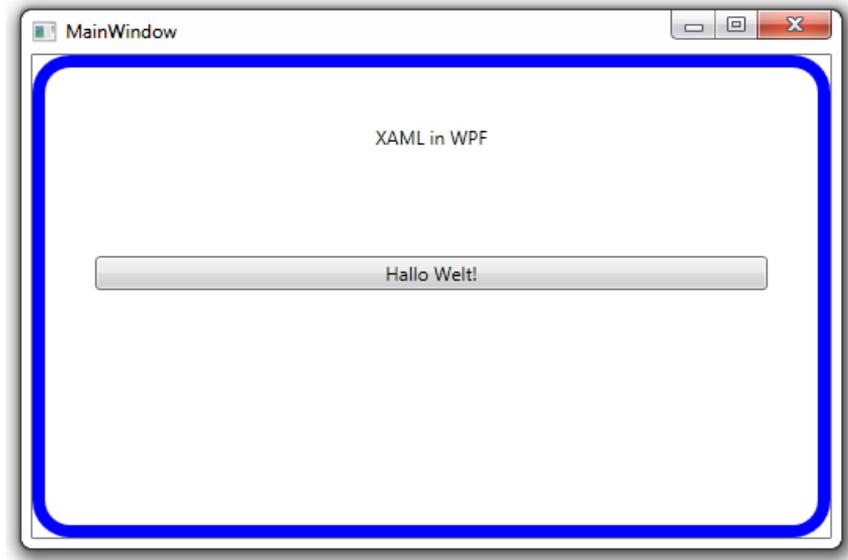
- Visual Studio 2008 / .NET 3.0 bis 3.5
 - WPF und Workflow verwenden jeweils eigene XAML Parser
- Visual Studio 2010 / .NET 4.0
 - XAML Services als gemeinsamer Parser
- Wichtige Klassen aus den XAML Services
 - XamlServices
 - XamlReader
 - XamlWriter
 - XamlSchemaContext

Demo

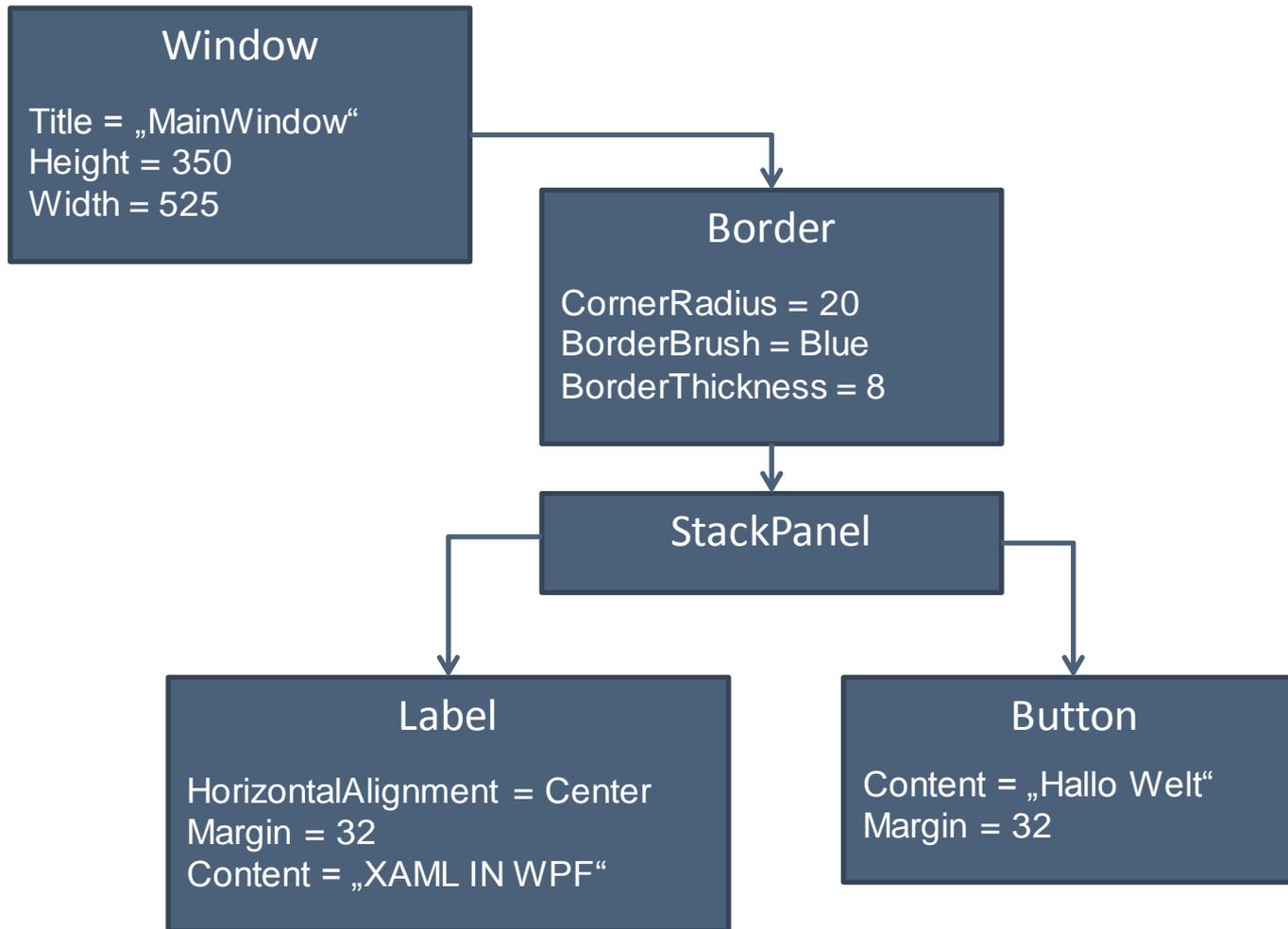
Eigene Typen als XAML serialisieren

Wie viele Objektinstanzen erzeugen wir hier?

```
1 <Window x:Class="WpfXamlDemo.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     Title="MainWindow" Height="350" Width="525">
5     <Border CornerRadius="20"
6         BorderBrush="Blue"
7         BorderThickness="8">
8         <StackPanel>
9             <Label HorizontalAlignment="Center" Margin="32">
10                XAML in WPF
11            </Label>
12            <Button Content="Hallo Welt!" Margin="32" />
13        </StackPanel>
14    </Border>
15 </Window>
```



Sieht der erzeugte Objektgraph so aus?



XAML als C# Code

```
1 <Window x:Class="WpfXamlDemo.MainWindow"
2     xmlns="http://schemas.microsoft.com/wi
3     xmlns:x="http://schemas.microsoft.com/
4     Title="MainWindow" Height="350" Width=
5 <Border CornerRadius="20"
6     BorderBrush="Blue"
7     BorderThickness="8">
8     <StackPanel>
9         <Label HorizontalAlignment="Center
10            XAML in WPF
11        </Label>
12        <Button Content="Hallo Welt!" Marg
13    </StackPanel>
14 </Border>
15 </Window>
```

```
var window = new Window();
window.BeginInit();
window.Title = "MainWindow";
window.Height = 350;
window.Width = 525;

var border = new Border();
border.BeginInit();
border.CornerRadius = new CornerRadius(20);
border.BorderBrush = new SolidColorBrush(Colors.Blue);
border.BorderThickness = new Thickness(8);

var panel = new StackPanel();
panel.BeginInit();

var label = new Label();
label.BeginInit();
label.HorizontalAlignment = HorizontalAlignment.Center;
label.Margin = new Thickness(32);
label.Content = "XAML in WPF";
label.EndInit();
panel.Children.Add(label);

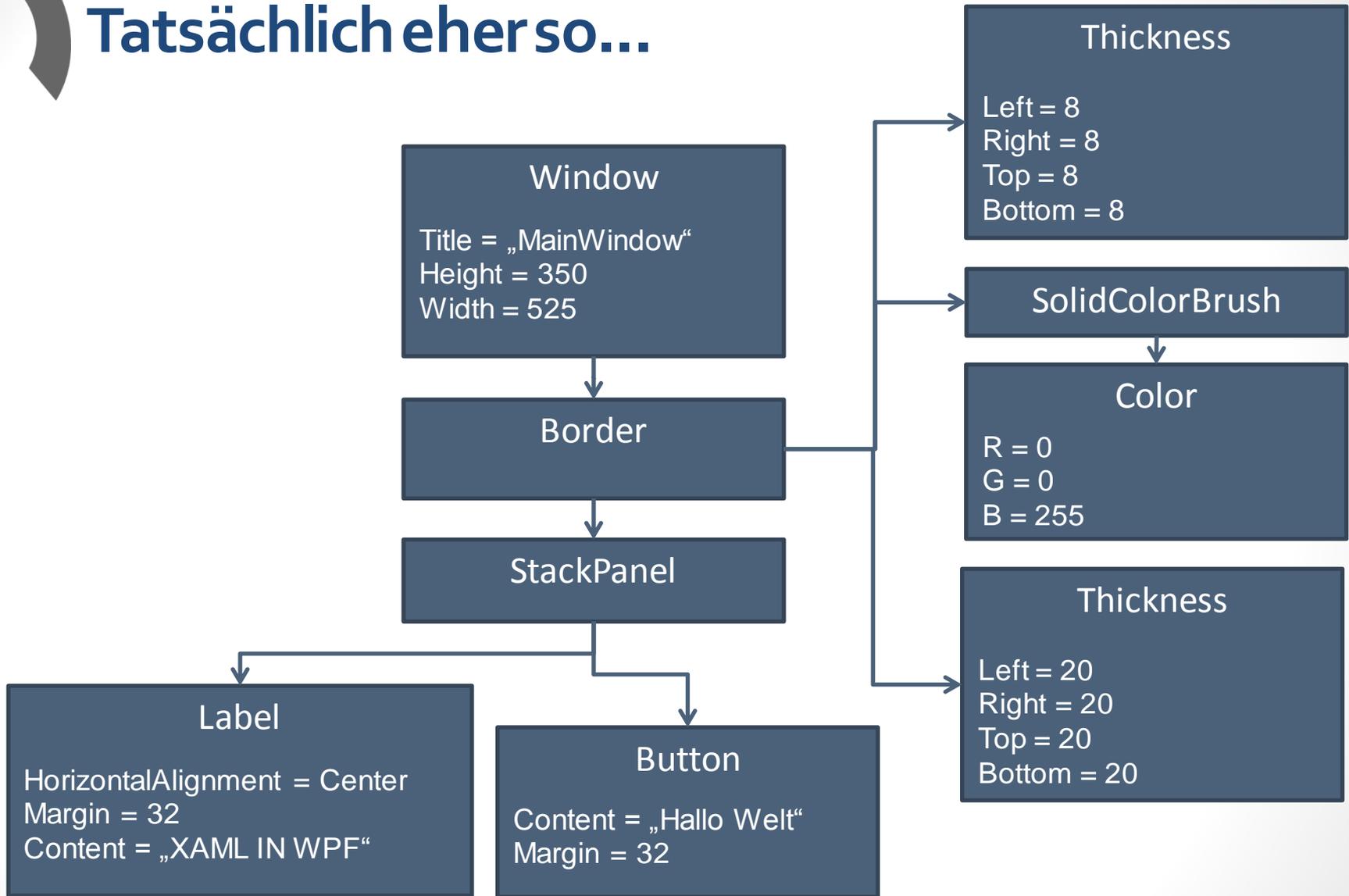
var button = new Button();
button.BeginInit();
button.Content = "Hallo Welt!";
button.Margin = new Thickness(32);
button.EndInit();
panel.Children.Add(button);

panel.EndInit();

border.Child = panel;
border.EndInit();

window.Content = border;
window.EndInit();
```

Tatsächlich eher so...



Demo

TypeConverter und Markupextensions

Demo

Attached Properties

Attached Properties

Beispiele

- DockPanel.Dock
- Grid.Row

Auch in „allgemeinem“ XAML Verwendbar

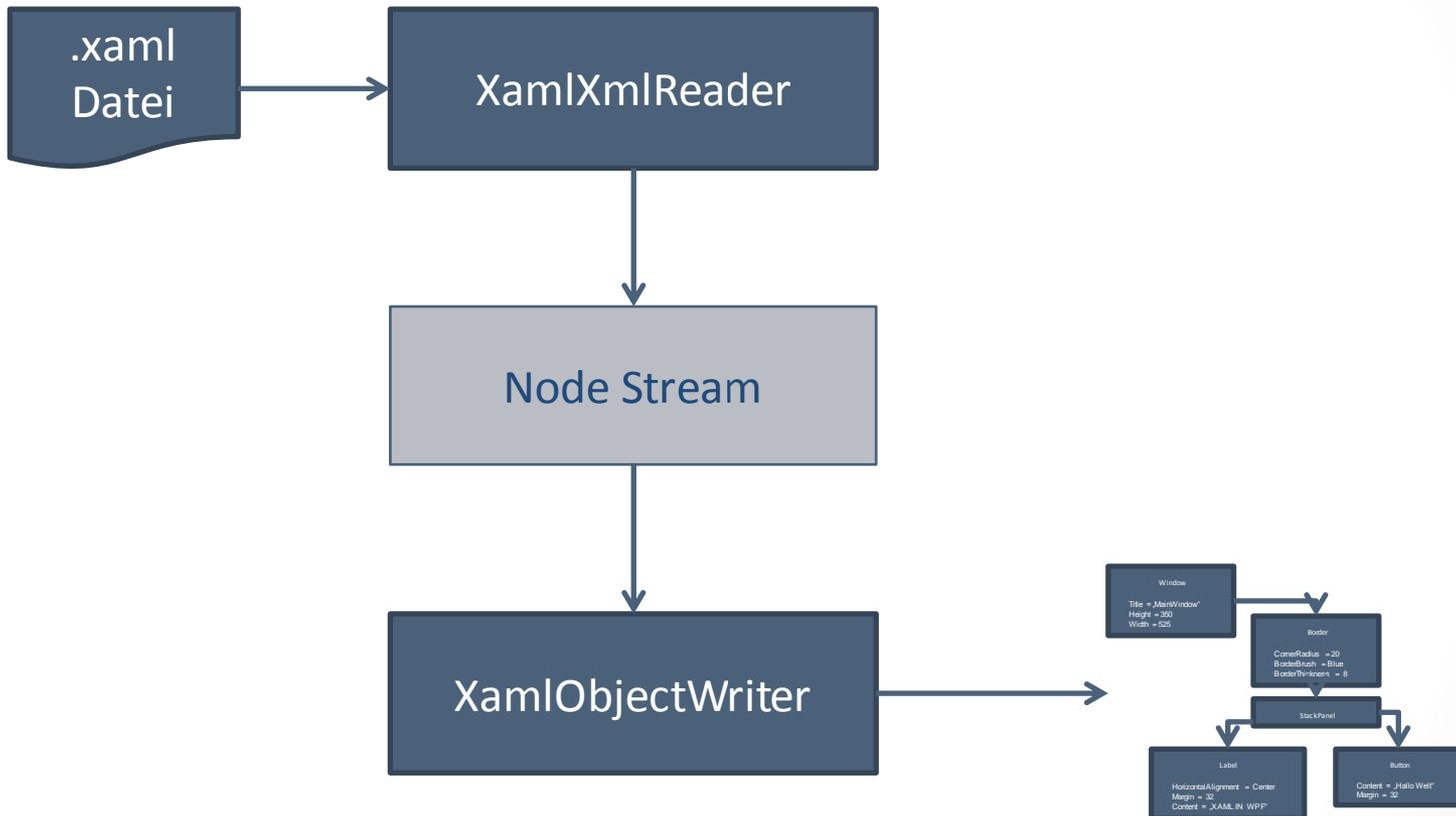
Werden erstellt über statische Methoden

- `GetEigenschaftName(object ziel)`
- `SetEigenschaftName(object ziel, object wert)`

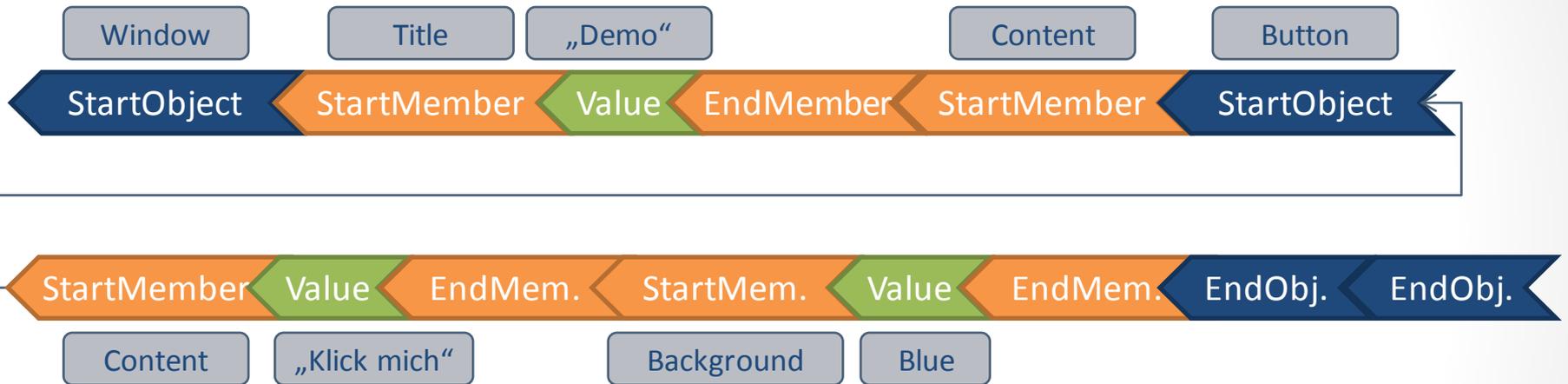
Serialisierung

- `IAttachedPropertyStore`

Von XAML Code zu Objektinstanzen



Der XAML Node Stream



Demo

Der XAML NodeStream

XAML Schema Context

- Zuordnung von Elementen und Namespaces zu Klassen und Assemblies
- WPF
 - `System.Windows.Markup.XamlReader.GetWpfSchemaContext();`
- Silverlight
 - XAML Toolkit
 - <http://archive.msdn.microsoft.com/XAML/>
 - CTP, Seit ca. 1 Jahr keine Updates mehr

Demo

Ein einfacher XAML Editor

XAML 2009

Neue Features

- Konstruktoren mit Parametern
- Factory-Methoden
- Vereinfachte Verwendung von Standarddatentypen
- Referenzen (x:Reference)

Einschränkungen

- Buildprozess für WPF verwendet noch BAML2006 Parser
- Somit nur für nichtkompilierte XAML Dateien
- Keine Designerunterstützung in VS oder Blend

Demo

XAML 2009

XAML allgemein vs. WPF XAML

XAML allgemein

- Markuperweiterungen, z.B.
 - Binding
 - Static
 - Type
- Attached Properties

WPF XAML Implementierung

- DependencyObject
- Dependency Properties
- Property Metadaten



Vo
Nu
Ju

-
-
-

The screenshot shows the ILSpy application window. The left pane displays a tree view of the assembly's structure, with `Windows.UI.DirectUI.Markup.XamlReader` selected. The right pane shows the source code for `XamlReader`, which is a sealed class implementing `IXamlReader`. The code includes several using statements and a static method `Load`.

```
using System;
using System.Runtime.InteropServices;
using Windows.Foundation;
namespace Windows.UI.DirectUI.Markup
{
    [WebHostHidden, Version(100794368u), Static(typeof(IXamlReader))]
    public sealed class XamlReader : IXamlReader
    {
        public static extern IDependencyObject Load([In]
    }
}
```

Quelle: <http://stevenhollidge.blogspot.com/2011/06/xaml-in-windows-8.html>

) Fragen?

- raacke@nevelop.de
- <http://www.nevelop.de>
- <http://www.outofcoffeeexception.de>
- @nlocalize
- @oocx

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Mathias Raacke

Nevelop GmbH



NLocalize



Spezialisiert auf .NET

Statt viele Plattformen halbherzig zu unterstützen haben wir NLocalize für aktuelle .NET UI Technologien optimiert.



Visual Studio Integration

Lokalisieren Sie Ihre Anwendungen mit einem Tool das Sie kennen und jeden Tag benutzen – Visual Studio 2010.



Visuelle Lokalisierung

NLocalize unterstützt Sie mit einer visuellen Vorschau der Übersetzung, auch bei komplexen Benutzeroberflächen.



Ideal für agile Entwicklung

Lokalisieren Sie Ihre Anwendung bereits parallel zur Entwicklung, statt erst danach.



Anpassbar

Mit unserem SDK können Sie NLocalize anpassen und erweitern, z.B. um eigene Dateiformate oder Translation Memories.



Benutzerfreundlich

NLocalize ist einfach bedienbar und leicht erlernbar. Sie werden nicht von dutzenden Icons und Toolbars erschlagen.