

## Windows Presentation Foundation

Björn Friedrich

04.12.2008

Bachelorprojektseminar



- Ein Überblick
- Architektur der WPF
- Programmiermodell
- Konzepte
  - XAML
  - Controls & Layout
  - Dependency-Properties
  - Routed-Events
  - Ressourcen
  - Styles, Trigger und Templates
  - Daten
- Was es sonst noch so gibt...
- Quellen und Literatur

## Windows Presentation Foundation (WPF)

- *Programmiermodell* für Windows- & Web-Anwendungen
- Gegen Ende 2006 zusammen mit WCF, WF und WCS eingeführt
- Version WPF = Version .NET Framework



(Jens Ellerweg: [www.studentprogram.de/paderborn](http://www.studentprogram.de/paderborn))

# Ein Überblick

## *Warum WPF?*

4

### **Vorabversion der CTPs**

- scheinbar komplexe Aufgaben einfach und schnell umsetzbar
- Umfangreichere Anwendungen → Ernüchterung der Entwickler

→ **Grund:**

Konzepte:

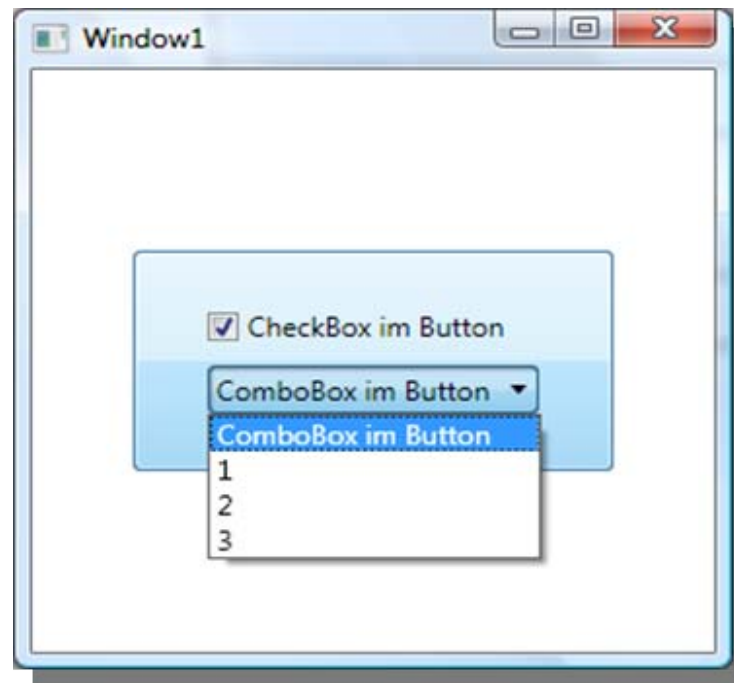
Layout  
Dependency-Properties  
Routed-Events  
Styles  
Templates

...

Meist unverstanden

## Flexibles Inhaltsmodell

- Verschachtelung von Elementen



# Ein Überblick

## *Vorteile und Stärken*

6

### **Styles**

- Meist als Ressource definierte *Sammlung von Eigenschaftswerten*
- Können Elementen zugewiesen werden

### **Trigger**

- Reagieren auf Eigenschaftsänderungen
- Deklarativ definiert

### **Lookless Controls**

- Visuelle Erscheinung und Logik von Controls getrennt
- *Controls selbst definieren nur Logik*
- Aussehen mittels `ControlTemplates` beschrieben

# Ein Überblick

## *Vorteile und Stärken*

7

### **Daten**

- Data Binding verknüpft Elemente mit verschiedensten Datenquellen
- Spart Programmieraufwand von Eventhandlern

### **2D- und 3D-Grafiken**

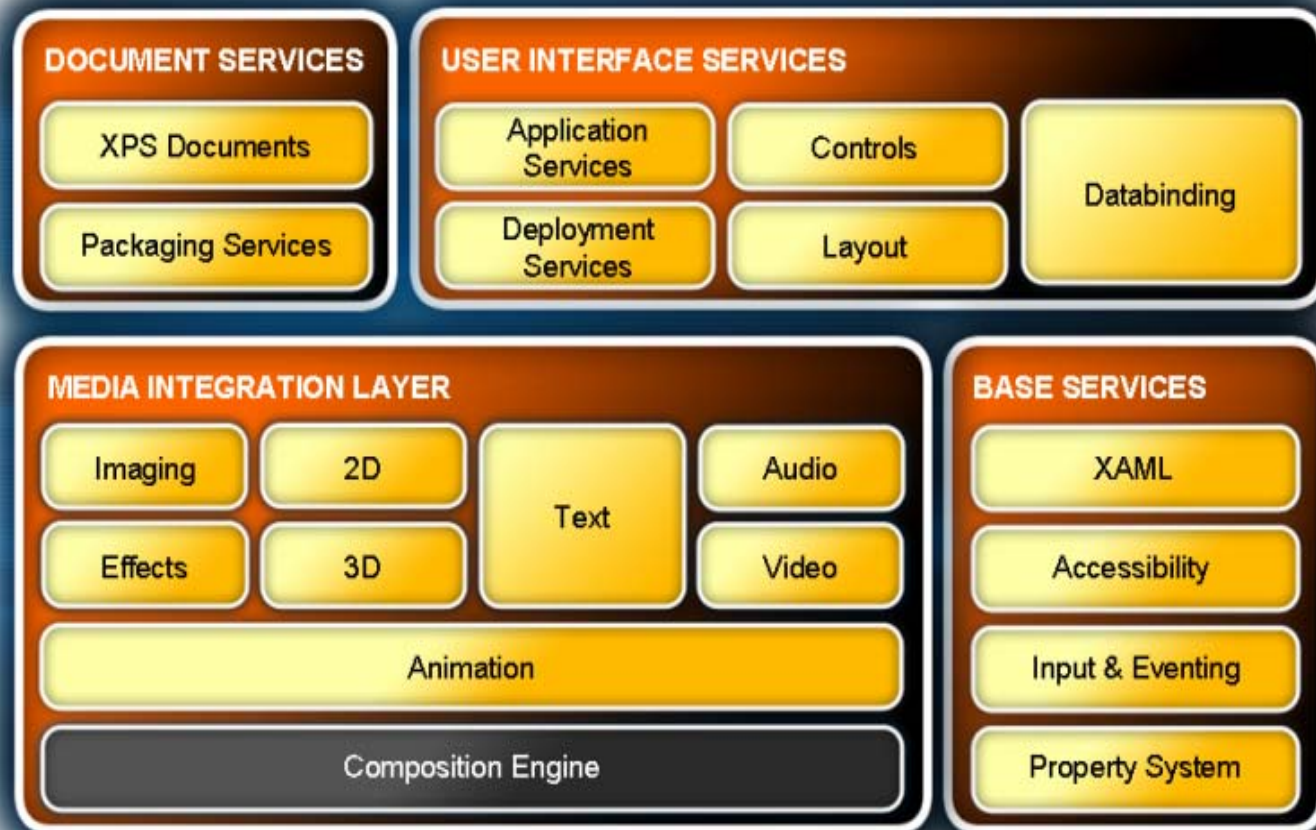
- Viele Zeichenwerkzeuge in der WPF vorhanden

### **Audio/ Video**

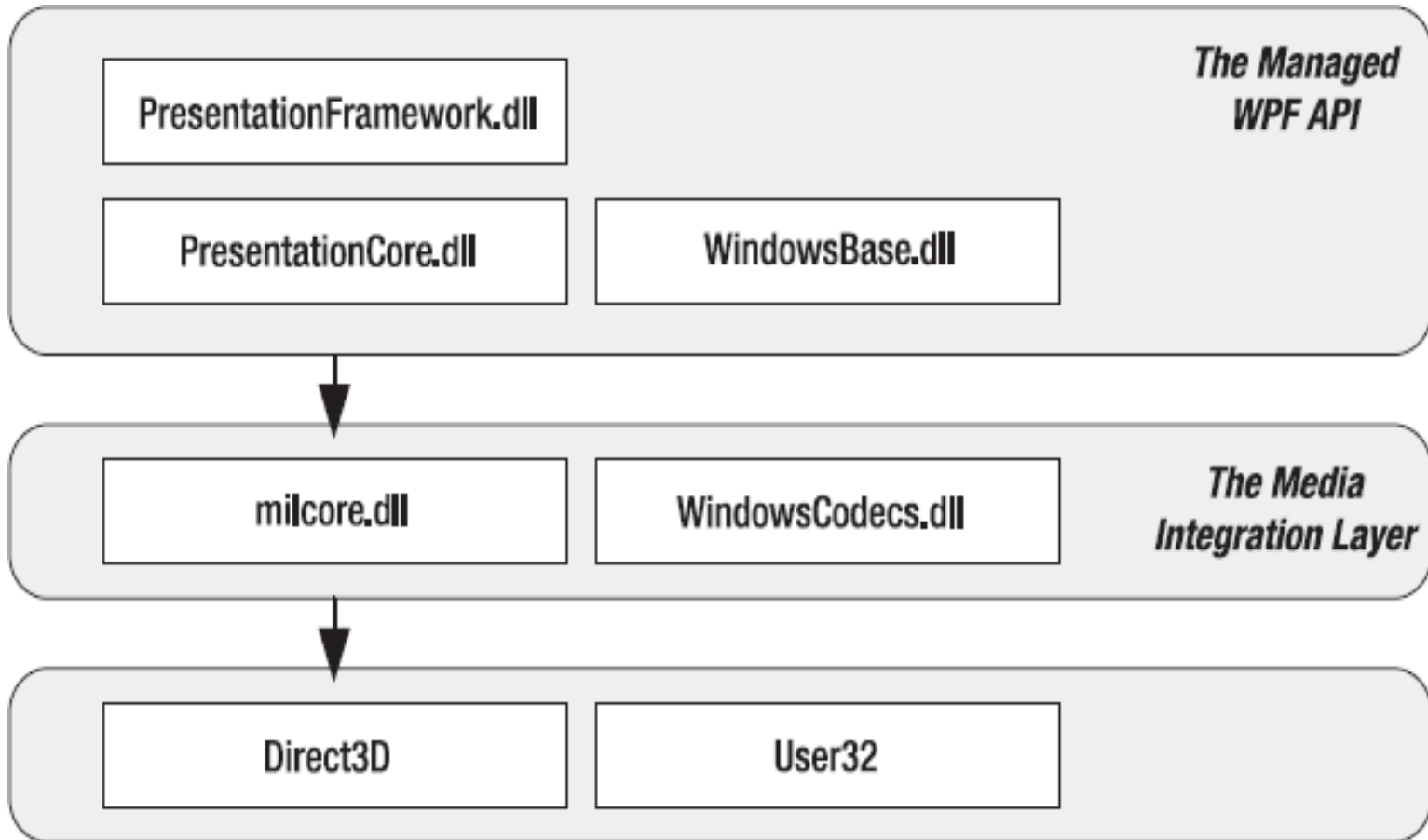
- Einfaches Einbinden von Audio/ Video

### **Text & Dokumente**

- Umfangreiche API zum Umgang mit Text, sowie fixen und fließenden Dokumenten



(Jens Ellerweg: [www.studentprogram.de/paderborn](http://www.studentprogram.de/paderborn))

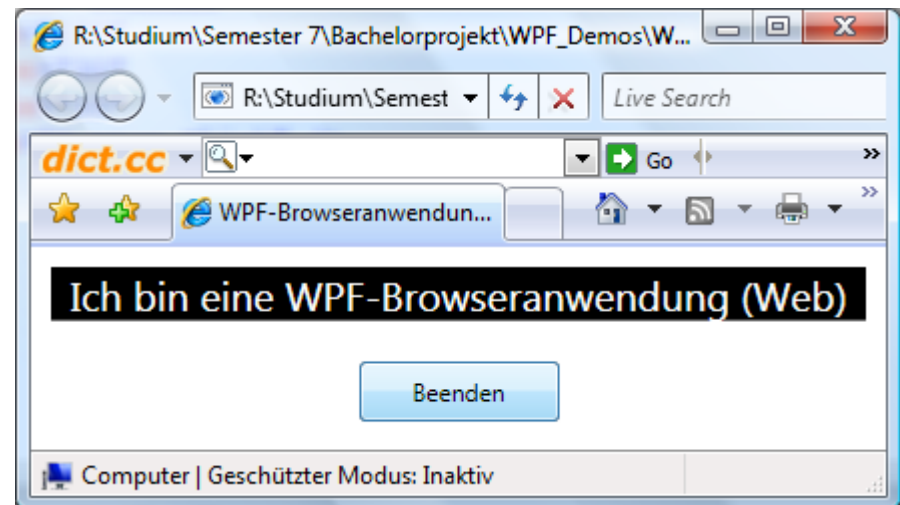
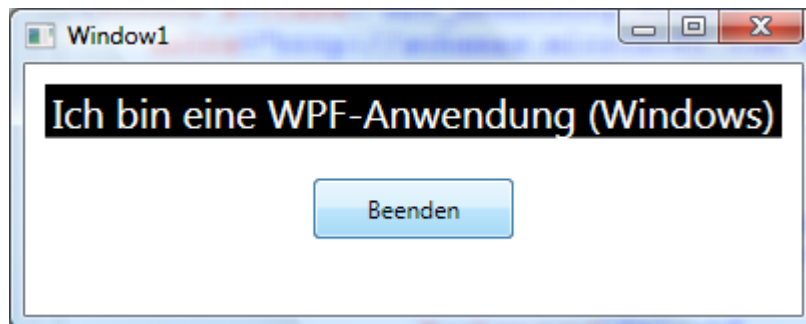


## WPF-Anwendungen (Windows)

- Wie WinForms
- Uneingeschränkte Rechte

## WPF-Browseranwendungen (Web)

- Auch XBAP (XAML Browser Application) genannt
- Sandbox: Internet Explorer → eingeschränkte Rechte
- Ab V3.5 auch in Firefox lauffähig



## ➤ XAML

- Controls & Layout
- Dependency-Properties
- Routed-Events
- Ressourcen
- Styles, Trigger und Templates
- Daten

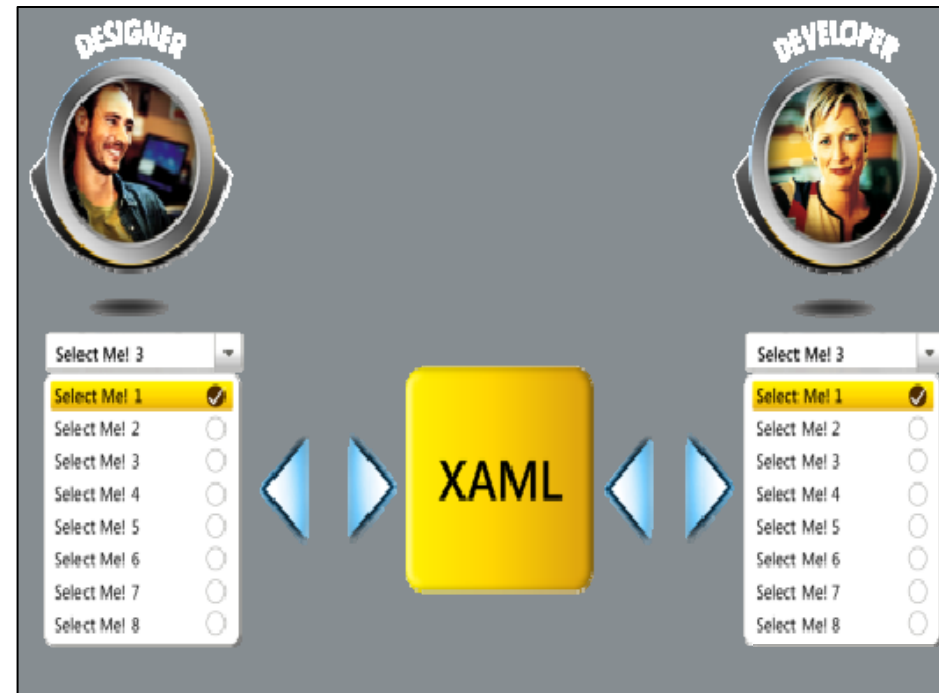


## eXtensible Application Markup Language

- XML-basierte Sprache zur Beschreibung von UIs
- Serialisierungsformat (Objekterzeugung aus XAML-Elementen zur Laufzeit)

### Trennung Darstellung—Logik

- XAML-Dateien für Designer
- Design-Tools
- Codebehind-Dateien für Entwickler



## Wesentlich kompaktere UI-Beschreibung

```
<Button Width="200" Height="30">Hallo</Button>
```

Äquivalenter C#-Code:

```
Button btnHallo = new Button();  
btnHallo.Width = 100;  
btnHallo.Height = 30;  
btnHallo.Content = "Hallo";
```



## XAML-Elemente

- Auch: *Objektelemente*

## XAML-Attribute

- Auch: *Property-Attribute* bzw. *Event-Attribute*

```
<Button Width="100" Height="30" Click="Button_Click" />
```

Objekt-  
element

Property-  
Attribute

Event-  
Attribut

## **XML-Namespace der WPF**

- In Assemblies hartkodiert
- *Mappt Objektelemente auf entspr. .NET-Klassen*

## **XML-Namespace für XAML**

- Spracherweiterungen/ Compiler-Direktiven (z.B. `x:Class`)

## **Namespace-Mapping**

- Erweitert XAML um eigene CLR-Namespaces

**DEMO 1**  
**REPEAT STRING**

### Type-Converters

- Konvertieren in Attributen angegebene Strings in das von der dahinter liegenden .NET-Property benötigte Objekt

```
<TextBlock Margin="5,8" />
```

**ThicknessConverter**



```
tb.Margin = new Thickness(5, 8, 5, 8);
```

- XAML dadurch kompakter

**DEMO 2**  
**TYPECONVERTER**

## Markup-Extensions

- Ableitungen von `System.Windows.Markup.MarkupExtension`
- *Kurznotation*
- In XAML in geschweiften Klammern

```
<ProgressBar
    Height="{x:Static Member=SystemParameters.IconHeight}"
    Minimum="{Binding ElementName=slProzent, Path=Minimum}"
    Maximum="{Binding ElementName=slProzent, Path=Maximum}"
    Value="{Binding ElementName=slProzent, Path=Value}" />
<Slider
    x:Name="slProzent"
    Height="{x:Static SystemParameters.IconHeight}"
    Minimum="0"
    Maximum="100"
    Value="50" />
```

### **Weitere MEs von XAML**

`x:Array`

`x:Null`

`x:Static`

`x:Type`

...

### **Weitere MEs der WPF**

`Binding`

`StaticResource`

`DynamicResource`

`ThemeDictionary`

...

## **XAML-Spracherweiterungen**

- Werden von Parser und Compiler verwendet

`x:Class`

`x:Key`

`x>Name`

`x:Code`

`xml:space`

`xml:lang`

...

- XAML
- **Controls & Layout**
- Dependency-Properties
- Routed-Events
- Ressourcen
- Styles, Trigger und Templates
- Daten

## Layout-Panels

- Grid
- StackPanel
- DockPanel
- Canvas
- ...

## ContentControls

- Buttons
- Labels
- Check Boxen
- Expander
- ...

## ItemsControls

- Menüs
- Toolbars
- Statusbars
- List Boxen
- Combo Boxen
- ...

## Sonstige

- Slider
- MediaElement
- ToolTip
- DocumentViewer
- ...

**DEMO 3**  
**LAYOUTPANELS**

- XAML
- Controls & Layout
- **Dependency-Properties**
- Routed-Events
- Ressourcen
- Styles, Trigger und Templates
- Daten

## Dependency-Properties

- *Grundlage für Styles, Animationen, Data Binding etc.*
- Voraussetzung für Setzen von Properties in XAML
- Integrierter Benachrichtigungsmechanismus
- Integrierte Validierung

## Property Engine

- `DependencyObject`: Oberklasse aller UI- und Framework-Elemente, die Property-Werte über Schlüssel intern in `Dictionary` ablegt
- `DependencyProperty`: definiert *Schlüssel* zum Wert einer Property

### DependencyObject

DependencyProperty	Value
Button.ContentProperty	Click Me
Grid.ColumnProperty	0
Grid.RowProperty	1
UIElement.IsEnabledProperty	True
...	...

GetValue

SetValue

**DEMO 4**  
**TEXTLABEL MIT FONTSIZE-  
PROPERTY**

- XAML
- Controls & Layout
- Dependency-Properties
- **Routed-Events**
- Ressourcen
- Styles, Trigger und Templates
- Daten

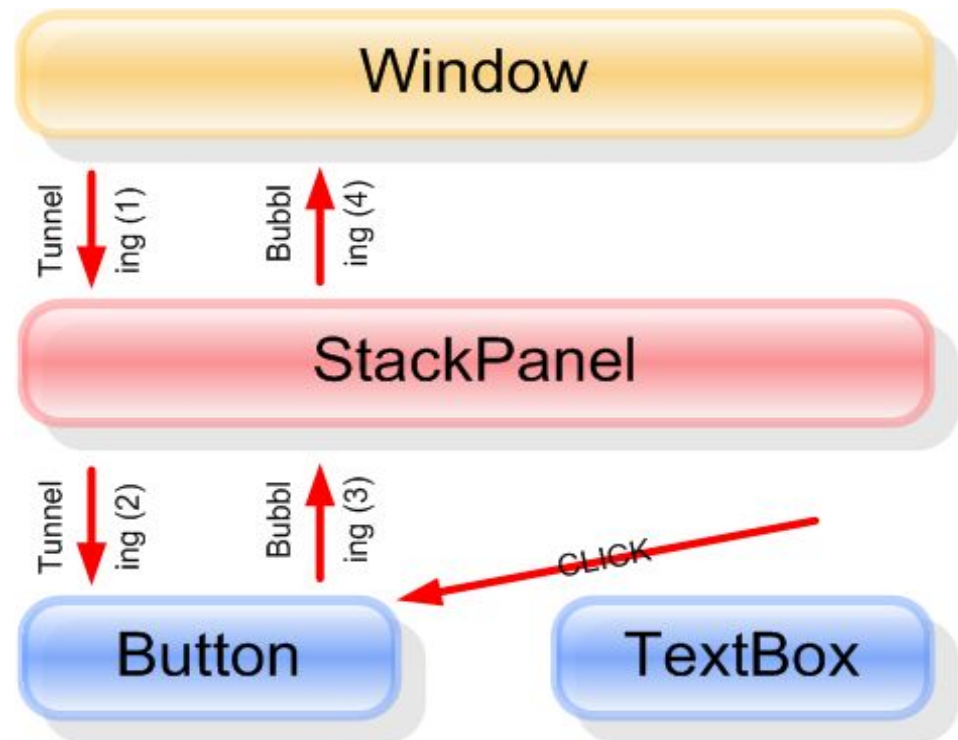
## Routed-Events

- Erlauben *Weiterleitung von Ereignissen* (z.B. `Button.Click`) an über- oder untergeordnete Elemente (z.B. `StackPanel`)

**Tunneling:** Weiterleitung von Events von oben nach unten im Element Tree

**Bubbling:** Events „blubbern“ von unten zu Elternelemente

**Direct:** Event wird nur auf dem event-auslösenden Element gefeuert



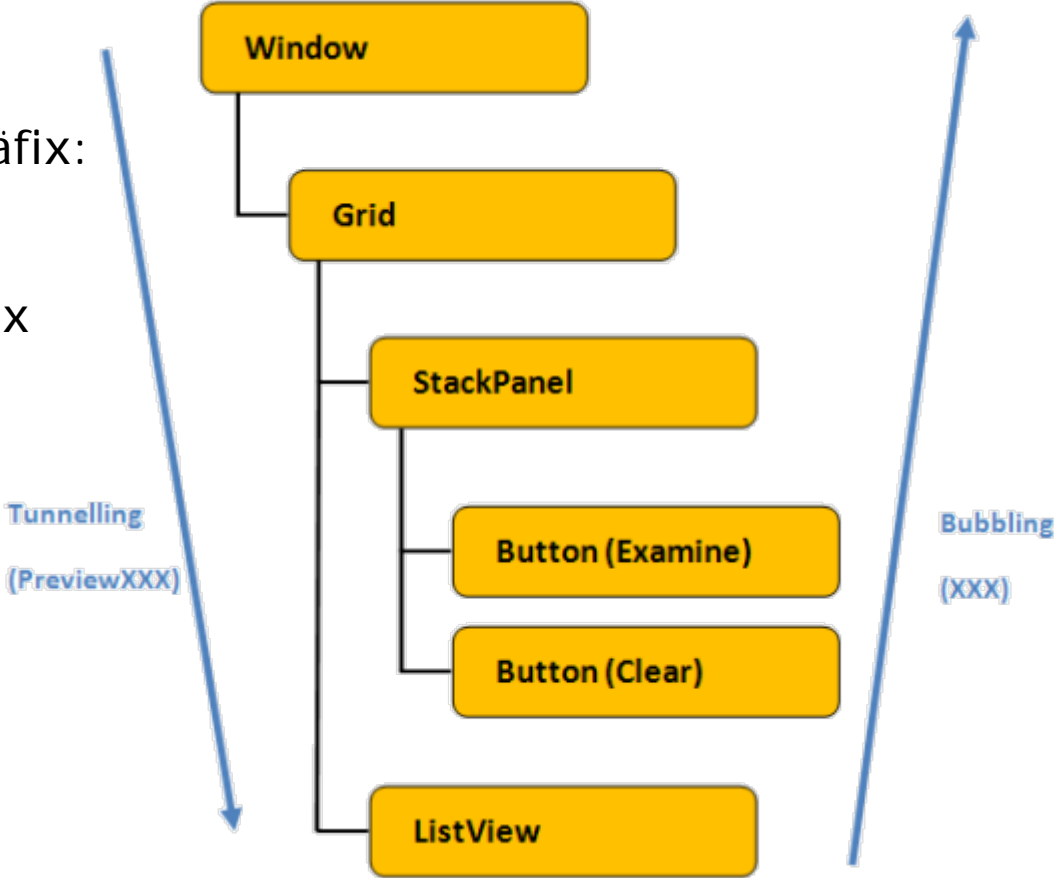
# Routed-Events

## Implementation und Anwendung

28

### Bezeichnungen

- Bubbling Events ohne Präfix:  
z.B. Click
- Tunneling Events mit Präfix  
Preview:  
z.B. PreviewClick



**DEMO 5**

**ROUTED EVENTS**

(MSDN)

- XAML
- Controls & Layout
- Dependency-Properties
- Routed-Events
- **Ressourcen**
- Styles, Trigger und Templates
- Daten

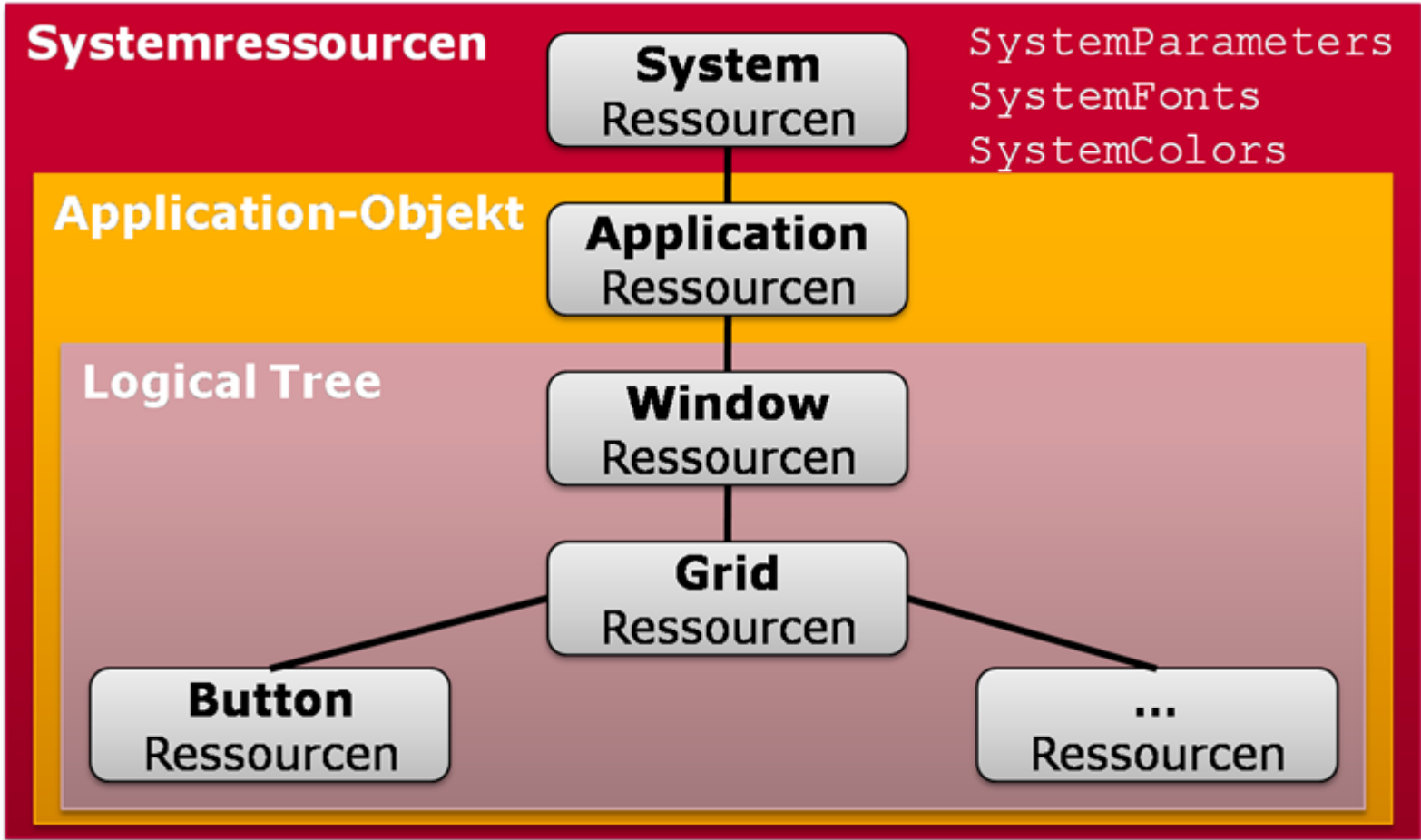
## Ressourcen

- Objekte, die zentral in `ResourceDictionary` definiert werden
- Können referenziert werden

## ResourceDictionary

- Speichert Schlüssel-/ Wertpaare
- Schlüssel mit `x:Key`-Attribut definiert

```
<StackPanel>
  <StackPanel.Resources>
    <SolidColorBrush x:Key="backgroundSolid"
                    Color="Orange" />
  </StackPanel.Resources>
  <Rectangle Fill="{StaticResource backgroundSolid}"
            Width="50"
            Height="50" />
</StackPanel>
```



### Statisch

- Property wird *einmalig* mit Ressource initialisiert

```
<Label  
    Content="{StaticResource {x:Static  
SystemParameters.PrimaryScreenWidthKey}}" />
```

### Dynamisch

- Ressource wird am Property „gebunden“
- Änderungen an Ressource → Änderung des Property-Wertes

```
<Label  
    Content="{DynamicResource {x:Static  
SystemParameters.PrimaryScreenWidthKey}}" />
```

**DEMO 6**  
**RESSOURCEN**

- XAML
- Controls & Layout
- Dependency-Properties
- Routed-Events
- Ressourcen
- **Styles, Trigger und Templates**
- Daten

### Styles

- Ermöglichen einfaches Anpassen von Control-Eigenschaften
- *Sammlung von Werten* für mehrere Dependency-Properties

```
<Button Content="Klick mich">
  <Button.Style>
    <Style>
      <Style.Setters>
        <Setter Property="Button.Background" Value="Black"/>
        <Setter Property="Button.Foreground" Value="White"/>
        <Setter Property="Button.Width" Value="100"/>
        <Setter Property="Button.FontSize" Value="20"/>
      </Style.Setters>
    </Style>
  </Button.Style>
</Button>
```

### Styles als logische Ressource

- **Benannte Styles:** müssen *explizit* über Markup-Extension `StaticResource` oder `DynamicResource` referenziert werden
- **Implizite Styles:** werden *implizit* von einem Element referenziert, wenn `x:Key-Attribut` oder `TargetType` des `Style` den Typ des Elements entspricht

**DEMO 7**  
**STYLES**

## Trigger

- Dynamische Aktionen definierbar
- Bestehen aus *Bedingung* und *Aktion*

## Trigger-Aktionen

- **EnterActions:** Collection von `TriggerAction`-Objekten, die ausgeführt werden, *wenn Trigger aktiviert wird*
- **ExitActions:** Collection von `TriggerAction`-Objekten, die ausgeführt werden, *wenn Trigger deaktiviert wird*

### Trigger-Arten

- **Trigger (Property-Trigger)**: wird ausgelöst sobald bestimmte *Dependency-Property* bestimmten Wert annimmt
- **DataTrigger**: wird ausgelöst, sobald bestimmte *.NET-Property*, die mittels Data Binding referenziert wird, bestimmten Wert annimmt
- **EventTrigger**: wird ausgelöst, sobald bestimmtes *Routed-Event* auftritt

**DEMO 8**  
**TRIGGER**

## **Templates**

- Beschreiben *Aussehen* von Controls oder Daten, indem sie den Visual Tree für ein Control bzw. die Visualisierung bestimmter Daten definieren

## Template-Arten

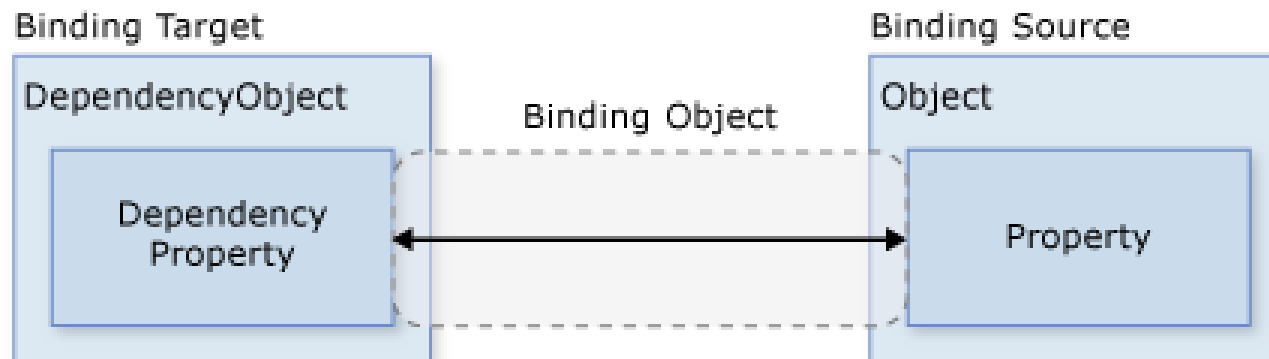
- **ControlTemplates:** beschreiben Aussehen von Controls, da Controls „lookless“ (definieren lediglich Logik)
- **DataTemplates:** definieren Aussehen für Daten
- **ItemsPanelTemplates:** definieren das Panel, das von einem ItemsControl (z.B. ListBox) für das Layout der Items verwendet wird

**DEMO 9**  
**TEMPLATES**

- XAML
- Controls & Layout
- Dependency-Properties
- Routed-Events
- Ressourcen
- Styles, Trigger und Templates
- **Daten**

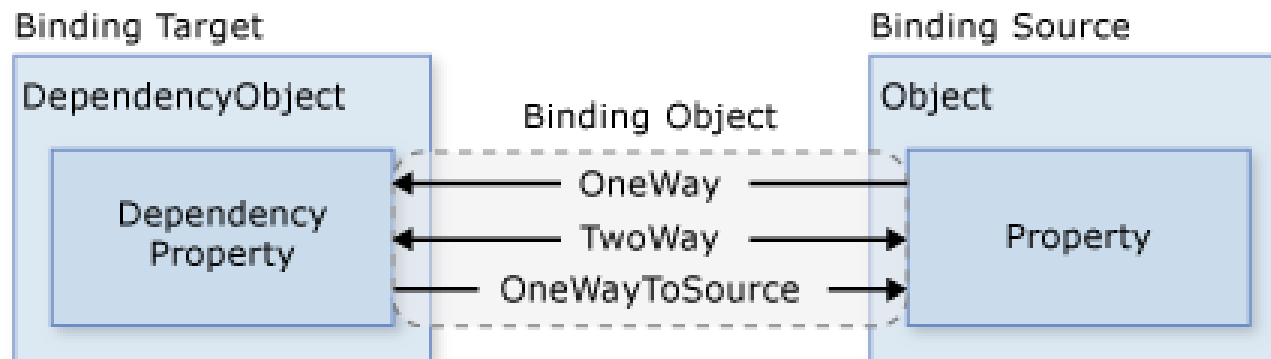
## Data Binding

- Bindet Wert einer Dependency-Property an
  1. Ressourcen (siehe `StaticResource`/ `DynamicResource`)
  2. Wert einer anderen Property (Binding-Klasse)



(MSDN)

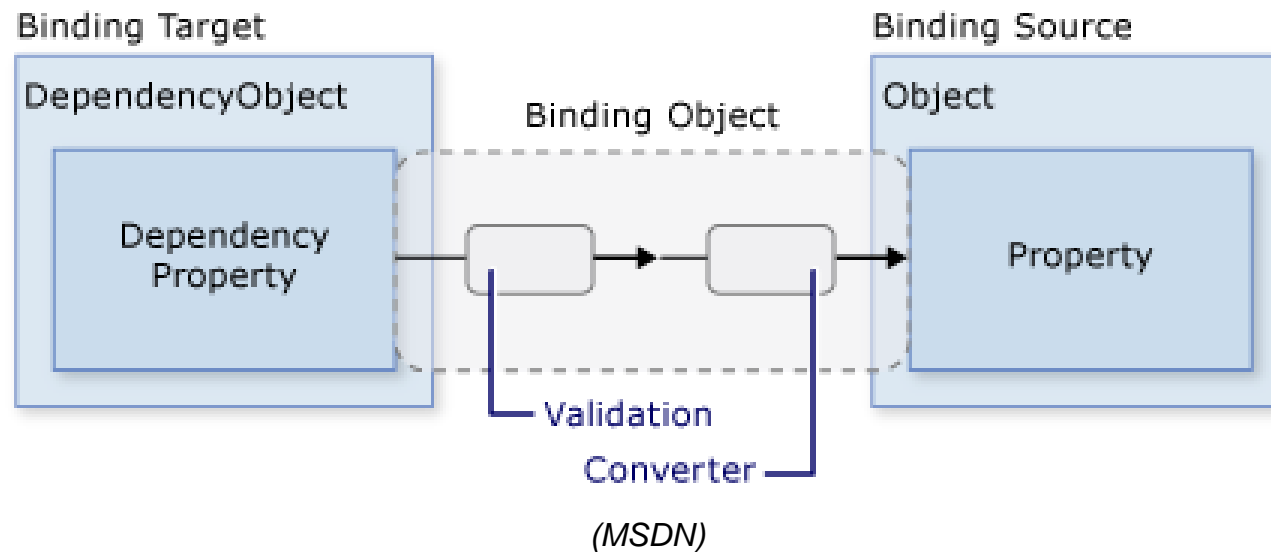
## Richtung des Data Bindings



(MSDN)

## Weitere Möglichkeiten

- Sortieren
- Filtern
- Wertkonvertierung und Validierung



**DEMO 10**  
**DATA BINDING**

- XAML
- Controls & Layout
- Dependency-Properties
- Routed-Events
- Ressourcen
- Styles, Trigger und Templates
- Daten

**Was es sonst noch so gibt...**

- Grafik
- Animationen
- Audio/ Video
- Text und Dokumente
- Win32-Interop
- ...

**EXTRA-DEMO**  
**SPIELEREIEN**

# Quellen und Literatur

**MSDN:** <http://msdn.microsoft.com>



**Code Project:** [www.codeproject.com](http://www.codeproject.com)



**Kotz et.al.:** *.NET 3.5 WPF, WCF und WF.*  
Addison-Wesley (2007)



**Huber:** *Windows Presentation Foundation.*  
Galileo Computing (2008)

