



Thomas Claudius Huber | Trivadis AG

WPF 4.0 – die Neuerungen

Über Thomas Claudius Huber

- .NET Senior Consultant
 - Trivadis AG Basel
 - Fokus: .NET, WPF, Silverlight, SqlServer, Oracle
- Autor der „umfassenden Handbücher“ zur WPF und zu Silverlight



- www.thomasclaudiushuber.com

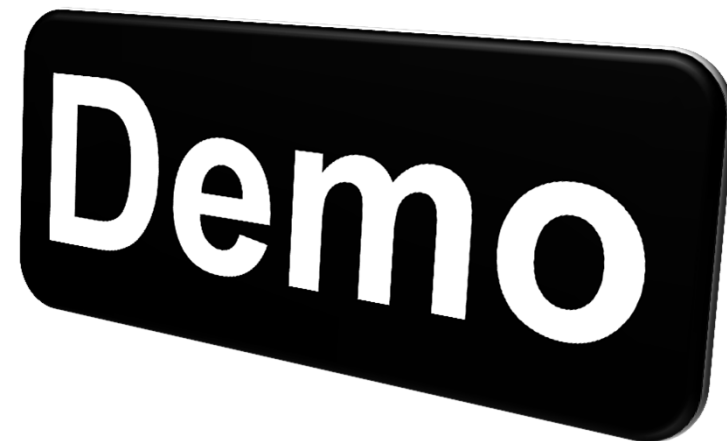
Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Neue Controls

- WPF enthält in der 4.0-Version drei neue Controls
 - DataGrid
 - Calendar
 - DatePicker
- Microsoft verspricht fast 99% Kompatibilität mit jenen Controls aus Silverlight



Demo

Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Visual State Manager - Controls

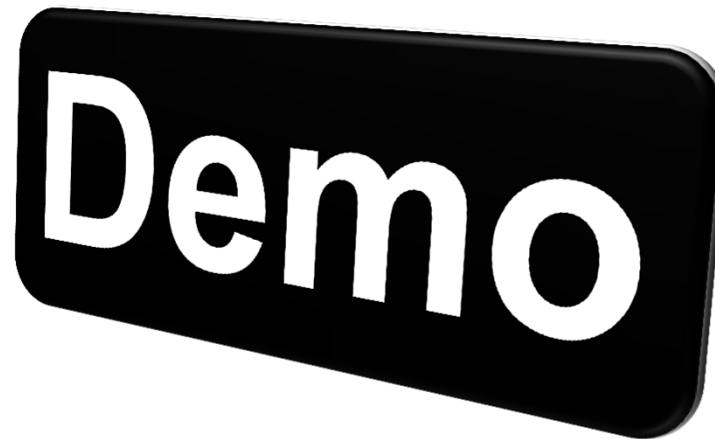
- WPF Controls sind als „lookless“ Controls implementiert – das Aussehen ist von der Logik getrennt
- Das Aussehen eines WPF Controls wird über ein ControlTemplate (XAML) definiert
- Die Logik wird klassisch in C# oder VB.NET implementiert

Visual State Manager – Verbindung

- Zum Verbinden von ControlTemplate und Logik enthält WPF das „Parts und States“-Modell
 - Parts definieren erwartete Teile eines Templates
 - States (neu in WPF 4.0) definieren die Zustände
 - Transitions
 - StateGroups
- Parts sind nichts neues, schon in WPF 3.0 vorhanden
 - Sind benannte Elemente im ControlTemplate
- States sind neu und wurden von Silverlight übernommen
 - Die ControlTemplates der WPF verwenden weiterhin Trigger, unterstützen allerdings die Visual States

Visual State Manager - Parts

- Benannte Elemente im ControlTemplate
- Auf Klassenebene mit dem TemplatePart-Attribut beschrieben
 - Die ProgressBar definiert die Parts *PART_Track* und *PART_Indicator*



Visual State Manager – Visual States

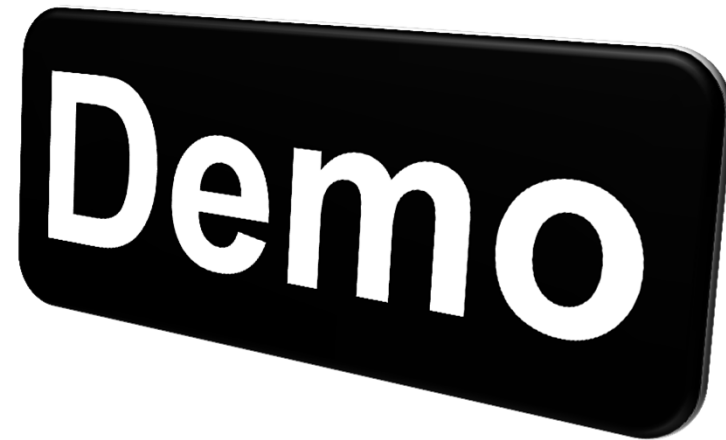
- Ein Visual State beschreibt einen visuellen Zustand eines Controls
- Visual States lassen sich gruppieren
 - Aus jeder Gruppe ist genau ein Visual State aktiv
- Visual States werden auf Klassenebene mit dem `TemplateVisualState-Attribute` deklariert.

Visual State Manager – Visual States

- Die Klasse `VisualStateManager` definiert die Attached-Property `VisualStateGroups`
 - Diese wird in einem `ControlTemplate` gesetzt, um die `VisualState`-Elemente und die darin beinhalteten Animationen zu definieren
- Neben den `VisualState`-Elementen lassen sich auch sogenannte `Transitions` zu einer `VisualStateGroup` hinzufügen
 - Eine `Transition` ist eine einfache Animation für den Übergang von einem Zustand zum anderen
- Die statische `GoToState`-Methode der `VisualStateManager`-Klasse wird zum Wechseln in einen anderen Zustand genutzt

VisualStateManager – Visual States

- Expression Blend bietet visuelle Unterstützung für das States-Modell



Demo

Agenda

- Neue Controls
- Visual State Manager
- **Multitouch**
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Multitouch

- Die WPF-Klassen `UIElement`, `UIElement3D` und `ContentElement` besitzen Touch-Events
 - `TouchDown`, `TouchUp`
 - `TouchEnter`, `TouchLeave`
 - `TouchMove`
- Nicht verwechseln mit Mouse-Events
 - Die WPF unterscheidet Touch- und Mouse-Events
 - Ein Mausklick löst kein `TouchDown`-Event aus
- `UIElement` unterstützt Touch-Bearbeitung
 - Skalieren, Rotieren und Verschieben
 - `IsManipulationEnabled`-Property auf `true` setzen und „Manipulation...“-Events verwenden

Multitouch

- Tipp zum Ausprobieren: Neben kompletten Multitouch-Notebooks gibt es auch einfache Bildschirme zum Testen der Multitouch-Funktionalität
 - Sinnvoll, wenn ein Rechner mit Windows 7 schon vorhanden ist
- Diese Bildschirme lassen sich einfach per VGA/DVI und USB-Kabel an einen Windows 7 Rechner anschliessen und werden als Multitouch-Gerät erkannt.
- Die einzige preisgünstigere Alternative sind Emulatoren, die das Anschliessen und Arbeiten mit zwei Mäusen erlauben.

Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- **Grafik**
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Grafik

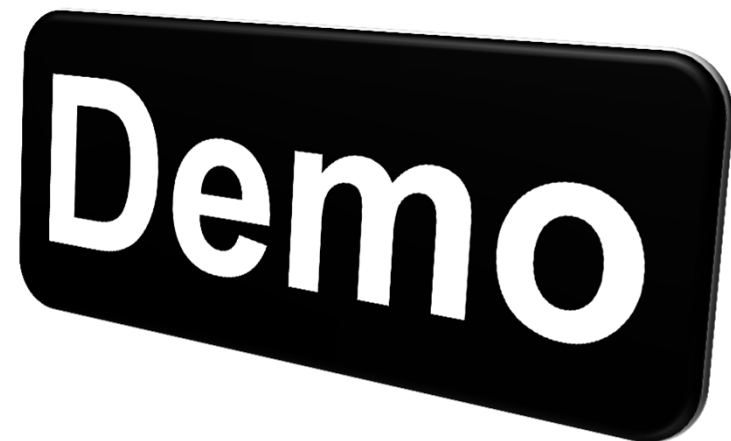
- Layout-Rounding
- Cached Compositions
- Unterstützung für Pixel Shader 3

Layout-Rounding

- Die Grösseneinheiten bei der WPF sind pixelunabhängig
 - Sogenannten logische Einheiten
 - Eine logische Einheit = $1 / 96$ inch
- Während dem Layout-Prozess können Elemente nicht genau auf eine Einheit und somit evtl. auch nicht genau auf einen Pixel fallen
 - Effekte sind halbtransparente oder verschwommene Kanten
- Layout-Rounding optimiert
 - Während dem Layout-Prozess (Measure und Arrange) werden alle nicht ganzzahligen Werte gerundet

Layout-Rounding

- FrameworkElement definiert die UseLayoutRounding-Property
 - Default-Wert ist false
- Die UseLayoutRounding-Property wird über den Element Tree vererbt und lässt sich somit einfach auf dem Wurzelement setzen



Cached Compositions

- Ein Teil des Visual Trees lässt sich als Bitmap zwischenspeichern
 - Dieser Teil muss beim erneuten Rendering nicht neu gezeichnet werden
- Performance wird dadurch verbessert
- Das Bitmap antwortet weiterhin auf Benutzereingaben

Cached Compositions

- Caching wird eingeschaltet, indem auf einem Element die CacheMode-Property auf den Wert BitmapCache gesetzt wird
- Achtung: Transformationen können eine schlechtere Qualität haben, da das Bitmap nicht neu gezeichnet wird
 - Somit kein Anti-Aliasing (=Glätten der Kanten)



Demo

Pixel Shader 3 Unterstützung

- Die in .NET 3.5 SP1 eingeführte ShaderEffect-Klasse wurde erweitert
 - Effekt wird der Effect-Property eines Elements zugewiesen
- Effektlibrary gibt es bereits für .NET 3.5 SP1
 - <http://www.codeplex.com/wpffx>
 - DirectX SDK wird dazu benötigt
- Die BitmapEffect-Property wurde als obsolete markiert
 - Aufgrund von Performance soll stattdessen die Effect-Property verwendet werden
 - Für die Effect-Property stehen standardmäßig die Effekte Blur und DropShadow zur Verfügung



Demo

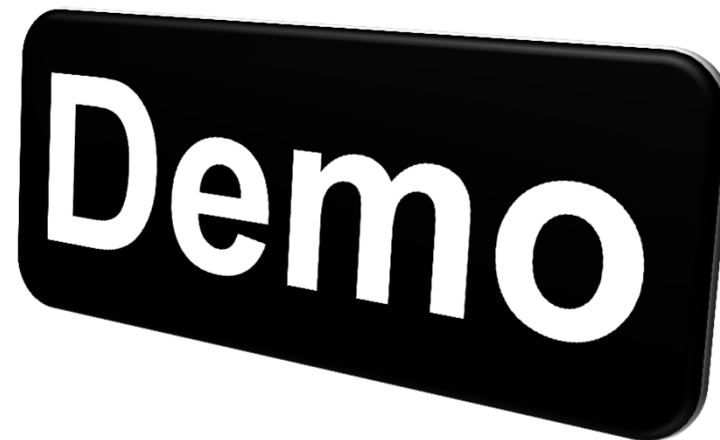
Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Animation Easing Functions

- Funktionen für Animationen
 - Bounce, Circle, Elastic usw.
- Wurden aus Silverlight übernommen
- Lassen sich in Expression Blend einfach via Designer nutzen



Demo

Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- **Binding**
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Binding

- Properties der InputBinding-Klassen wurden als Dependency-Property implementiert
 - MouseButton und KeyBinding existierten schon in .NET 3.5 SP1, die Properties waren allerdings nicht „bindable“
- Command-Property eines InputBindings lässt sich somit direkt an ein Command aus einem ViewModel binden



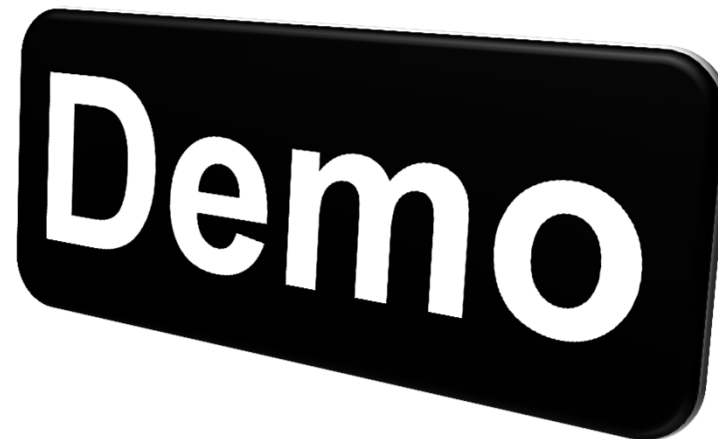
Demo

Binding Dynamic Objects

- WPF 4.0 unterstützt das Binden an dynamische Objekte
 - In XAML und im Code (C#/VB.NET)
- Dynamische Objekte wurden in .NET 4.0 eingeführt
 - dynamic-Keyword in C# für Late-Binding
- Eigene Klassen von DynamicObject ableiten oder IDynamicMetaObjectProvider implementieren

Text-Property des Run-Element

- Run-Element repräsentiert ein Stück Text in
 - einem TextBlock
 - einem FlowDocument
- Text-Property des Run-Elements wurde in .NET 4.0 als Dependency-Property implementiert
 - Ist somit jetzt „bindable“



Demo

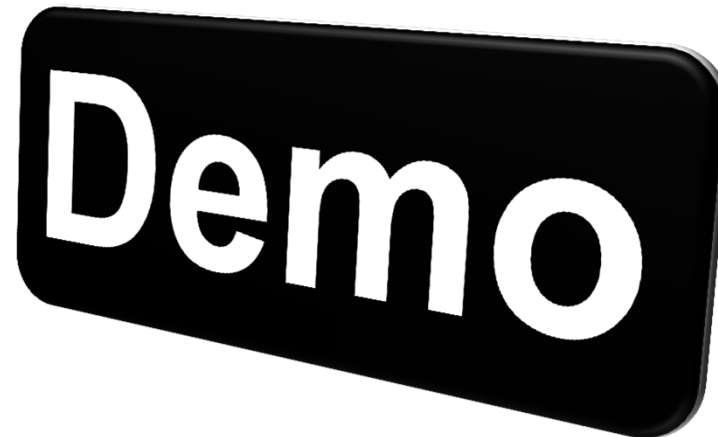
Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- **XBAP (XAML Browser Apps)**
- Windows 7 APIs
- XAML
- Visual Studio 2010



XBAP 4.0 Features

- HTML/XBAP-Interop
 - Über die statische HostScript-Property der BrowserInteropHelper-Klasse (System.Windows.Interop) gibt es das ScriptObject, das den Zugriff auf das HTML-Fenster ermöglicht
- Full-Trusted XBAPs
 - Installation erfordert Zustimmung des Benutzers



Demo

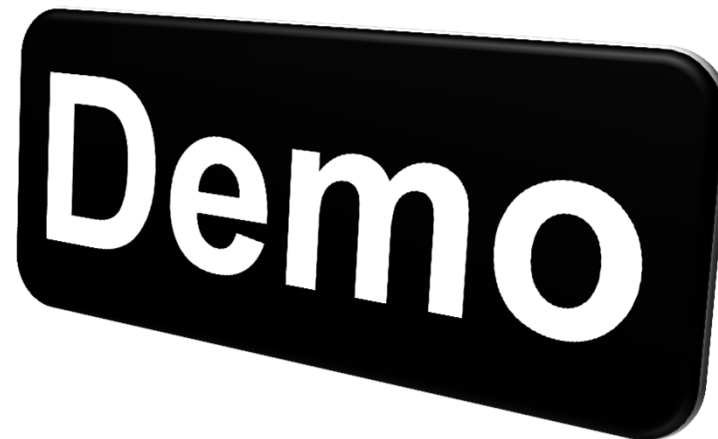
Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- **Windows 7 APIs**
- XAML
- Visual Studio 2010



Windows 7 APIs

- Im Namespace System.Windows.Shell befinden sich Klassen, um die TaskBar von Windows 7 zu steuern
- WPF-Anwendungen können die Funktionalität aus XAML und prozeduralem Code nutzen
- Läuft die Anwendung unter Windows < 7, macht die API einfach nichts
 - Es wird keine Exception geworfen



Demo

Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- **XAML**
- Visual Studio 2010



XAML Neuerungen

- XAML-Funktionalität wurde in die System.Xaml.dll gepackt
 - Eventuell verwenden WPF und Silverlight in Zukunft denselben XAML-Parser
- x:Reference Markup-Extension wurde eingeführt
- Zahlreiche neue Klassen, um im prozeduralen Code mit XAML zu arbeiten

XAML Zukunft

- Für XAML in .NET 4.0 waren weitere Features geplant, die es allerdings nicht in den Release schafften
 - Generics mit `x:TypeArguments` erstellen
 - Parametrisierte Konstruktoraufrufe mit `x:Arguments`
 - Factory-Methoden mit `x:FactoryMethod`

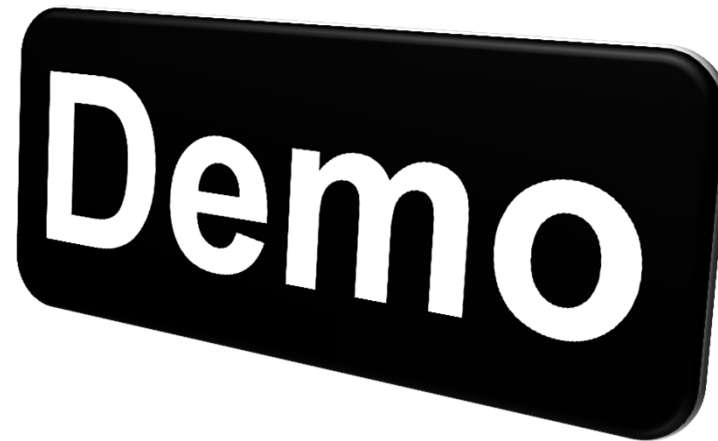
Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- **Visual Studio 2010**



Visual Studio 2010

- Data-Binding über Designer möglich
- Verbesserungen für automatisches Layout
- Zusätzliche Eigenschaften-Editoren
 - Farbpalette etc.
- Intellisense für XAML verbessert
 - Bspw. Für Markup-Extensions



Demo

Agenda

- Neue Controls
- Visual State Manager
- Multitouch
- Grafik
- Animation Easing Functions
- Binding
- XBAP (XAML Browser Apps)
- Windows 7 APIs
- XAML
- Visual Studio 2010



Fazit Neuerungen

- Insbesondere das DataGrid und der DatePicker sind für Business-Applikationen wichtig
 - Third-Party-Controls weiterhin für mehr Funktionalität notwendig
- Kompatibilität zu Silverlight wurde wieder etwas erhöht, dennoch gibt es reichlich Unterschiede
- Neben den gezeigten Änderungen gibt es weitere Verbesserungen
 - Bspw. Wurde der Text-Rendering-Stack komplett überarbeitet, wodurch WPF-genderter Text so scharf dargestellt wird wie GDI-genderter Text

Slides

- Slides sind heute Nachmittag auf meinem Blog online www.thomasclaudiushuber.com/blog
- Fragen, Kritik und Sonstiges direkt an thomas@thomasclaudiushuber.com

Besten Dank für Ihre Aufmerksamkeit!

Any Questions?!