

## .NET User Group Leipzig – .NET Bootcamp 2009: Silverlight / WPF

### Einfach

1. Erklären Sie, was die *Extensible Application Markup Language* (XAML) ist und warum sie für Silverlight / die WPF entwickelt wurde. Welche Bedeutung haben dabei Tags und deren Attribute? Zeigen Sie verschiedene Schreibweisen für das Setzen von Eigenschaften am Beispiel von *Brushes* (*SolidColorBrush* vs. *LinearGradientBrush*).
2. Welche Zusammenhänge gibt es zwischen dem Konzept „*Code Beside*“ und XAML? Erzeugen Sie eine Anwendung mit zwei Schaltflächen: Eine mit XAML, die andere mit C#. Kann man C#-Code auch in XAML einbetten? Wie kann man XAML dynamisch laden und im bestehenden XAML-Baum „einhängen“?
3. Was sind die elementaren Unterschiede zwischen Silverlight und der WPF? Gehen Sie auf Laufzeitumgebung, Crossplattformfähigkeit, Hardwareunterstützung (3D) und Unterschiede bei den Steuerelementen/Klassen (z. B. auch *System.Windows.Browser*) ein. Stellen Sie die Vor- und Nachteile, sowie Anwendungsgebiete dieser beiden Technologien gegenüber. Erläutern Sie kurz, was XBAP-Anwendungen sind.
4. Wie können Anwendungen sowohl für Silverlight als auch die WPF entwickelt werden? Gibt es hier bereits Lösungen oder Best Practices?
5. Was sind sogenannte *DependencyProperties* (Abhängigkeitseigenschaften)? Welche Unterschiede gibt es zu den „normalen“ .NET Properties? Welche Vorteile bieten *DependencyProperties* z. B. bei Datenbindung, Validierung usw.? Zusatz: Was sind „*angefügte Eigenschaften*“?
6. Aus welchen Komponenten besteht eine Silverlight-Anwendung? Zeigen Sie anhand eines mit Visual Studio neu erstellten Silverlight-Projektes, wie die Initialisierung beginnend mit dem Aufruf der Webseite im Browser abläuft.
7. Neben Visual Studio 2008 existieren weitere Werkzeuge für die Entwicklung von WPF und Silverlight-Anwendungen. Geben Sie einen Überblick und zeigen Sie kurz deren Verwendung (z. B. XAMLPad, WPF Performance Suite, Snoop, DeepZoom...).

## Mittel

8. Erzeugen Sie mit Hilfe von *Expression Blend* eine WPF-Animation einer Schaltfläche. Verwenden Sie dafür folgende Transformationen: *Translation*, *Rotation* und *Skalierung*. Erläutern Sie den dabei generierten XAML-Code. Zusatz: Wie kann man die Schaltfläche mit der Hilfe eines Pfades animieren? Lassen sich auch Farben und Farbverläufe animieren?
9. Welche *Inhaltsmodelle* für Steuerelemente gibt es in WPF? Geben Sie einen Überblick über die verschiedenen *Panel*-Klassen aus der WPF. Erstellen Sie einfache Beispiele, die den jeweiligen Container in seiner Anwendung demonstrieren. Welche Unterschiede gibt es bei den Panel-Klassen zu Silverlight? Tipp: Finden Sie Steuerelementbibliotheken, die mögliche Lücken schließen. Zusatz: Welche Parallelen beim Umgang mit WPF- / Silverlight Steuerelementen gibt es zu ASP.NET.
10. Welche Bedeutung haben *Trigger* für die WPF und Silverlight? Welche Arten von Triggern gibt es in der WPF und wo werden sie wie eingesetzt? Zeigen Sie für die WPF kurze Beispiele für deren jeweilige Anwendung. Welchen Unterschied gibt es zu Silverlight?
11. Erklären Sie *Markuperweiterungen* an zwei konkreten Beispielen für Ressourcen (z. B. Referenzierung eines *Brushes*) und *DataBinding* (z. B. *Element-To-Element Binding*). Zeigen Sie eine andere Schreibweise der Markuperweiterungen. Zusatz: Worin liegt der Unterschied zwischen *DynamicResource* und *StaticResource*? Gibt es *DynamicResource* auch in Silverlight?
12. Was sind *Routed Events*? Erklären Sie *Tunneling* und *Bubbling* anhand eines Beispiels: Zeigen Sie mithilfe des *MouseDown*-Ereignisses, welche Vorteile Routed Events in WPF gegenüber den Ereignissen in Windows Forms-Anwendungen haben. Welche Einschränkungen gibt es in Silverlight hinsichtlich der Routed Events?
13. Was ist ein *ResourceDictionary* in WPF / Silverlight? Worin unterscheiden sich dessen Ressourcen zu denen aus klassischen .NET Projekten (.resx)? Wo und wie können *ResourceDictionaries* angelegt bzw. verwendet werden? Welche Unterstützung bietet *Expression Blend* dafür?
14. Zeigen Sie, wie eigene Steuerelemente in XAML erstellt, referenziert und verwendet werden. Erstellen Sie hierzu ein kleines, übersichtliches WPF Projekt mit einem *User Control*, um z. B. folgende Datenstruktur darstellen zu können:



Zusatz: Zeigen Sie an diesem Beispiel die Verwendung von *DependencyProperties* (z. B. Validierung der *ImagePath* Abhängigkeitseigenschaft als gültige URI).

## Herausforderung

15. Erläutern Sie kurz die drei Arten des *Data Binding* (*Element-To-Element*-, *CLR*-, *XML-DataBinding*) und deren Update-Modus, sowie *UpdateSourceTrigger*. Ihre Aufgabe: Ein Slider (Werte 1..100) soll an eine TextBox gebunden werden. Der Wert (Value) des Sliders wird bidirektional an die Text-Eigenschaft der TextBox gebunden, so dass Aktualisierungen in beiden Richtungen erfolgen. Zusatz: Mithilfe eines *Konverters* (*IValueConverter*) soll die Hintergrundfarbe der TextBox entsprechend des eingestellten Wertes aktualisiert werden. (1..30: grün, 30..70: gelb, 70..100: rot).
16. Worin besteht der Unterschied zwischen *Styles* und *Templates*? Erläutern Sie unter Zuhilfenahme von *Expression Blend*, wie *Styles* und *Templates* erstellt und verändert werden können. Zusatz: Welche Rolle spielen die s. g. *DataTemplates*? Was ist der *Visual State Manager*?
17. Zeigen Sie an einem Beispiel, wie *ResourceDictionaries* dynamisch geladen werden können. Vorschlag: verschiedene Stile für Schaltflächen sind in separaten .xaml Dateien hinterlegt. Zur Laufzeit soll über eine „Laden“-Schaltfläche ein „Datei öffnen“-Dialog angezeigt werden mit dessen Hilfe es möglich ist die .xaml Dateien auszuwählen und den jeweiligen Stil entsprechend zu laden, so dass die Schaltflächen ihr Aussehen ändern.
18. Recherchieren Sie die verschiedenen Möglichkeiten WPF-Anwendungen zu *lokalisieren* / *globalisieren*. Zeigen Sie an einem Beispiel, die von Microsoft empfohlene Vorgehensweise. Gibt es Werkzeuge, die bei dieser Aufgabe unterstützen?
19. Welche *Unit Test Frameworks* unterstützen Silverlight? Zeigen Sie an einem konkreten Beispiel deren jeweilige Verwendung. Welche Vor- und Nachteile bieten die jeweiligen Frameworks? Warum können gängige Unit Test Frameworks nicht für Silverlight benutzt werden?
20. Finden Sie mögliche Grenzen der WPF hinsichtlich des Aspektes „*Leistungsfähigkeit*“. Was sollte man beachten, um performant agierende WPF / Silverlight Anwendungen zu entwickeln? Wie kann man mit der *WPF Performance Suite* „Flaschenhälsen“ entgegenwirken? Gehen Sie z. B. auf *Bitmapeffekte*, *DirectX-Unterstützung*, Verbesserungen im Service Pack 1 ein. Zusatz: Zeigen Sie die Verwendung der Klasse *CompositionTarget* im Vergleich zu *BackgroundWorker* oder „normalen“ WPF-Animationen.
21. Erstellen Sie einen einfachen *WCF-Webdienst* und einen entsprechenden Silverlight-Client, der diesen Dienst verwendet. Beispielvorschlag: Einen WCF-Webdienst erstellen, der Additions- und Subtraktionsoperationen bereitstellt oder einen bereits vorhandenen nutzen<sup>1</sup>. Der Silverlight-Client selber soll in Form eines Taschenrechners auf diesen Dienst zugreifen. Welche Probleme können hinsichtlich der *Sicherheit* auftreten? Zusatz: Zeigen Sie einen entsprechenden WPF-Client, der den gleichen Webdienst anspricht.

---

<sup>1</sup> <http://spielewiese.torstenweber.de/MathService/Operations.aspx>