



mitp

Uwe  
Rozanski

Microsoft

# Silverlight 4

# Was ist Silverlight?

## 1.1 Eigenschaften von Silverlight

Kurz und knapp gesagt, beschäftigt sich Silverlight mit der Erstellung grafischer Oberflächen für alle möglichen Arten von Anwendungen. Neben reinen Business-Applikationen, bei denen es vor allem auf Eingabefelder, Buttons, Subdialoge und so weiter ankommt, lassen sich auch Werbeauftritte designen, bei denen eher Wert auf knallige Animationen, Sounds, Videos und eine coole Benutzerführung gelegt wird. Silverlight legt sich da nicht fest, sondern lässt beides zu. Gerade die sinnvolle Kombination beider Präsentationsarten kann dem Anwender einen echten Mehrwert bringen. Denken wir zum Beispiel an eine Lagerverwaltungssoftware, bei der der verantwortliche Mitarbeiter über eine klassische ComboBox aus verschiedenen Livevideos auswählen kann, die ihm zeigen, welcher LKW gerade wie weit beladen ist und wie viele noch auf dem Parkplatz warten. Auch die Überwachung und Steuerung der Produktion kann so aus der Verwaltung heraus viel einfacher funktionieren und somit kann viel schneller auf individuelle Kundenwünsche eingegangen werden.

Die entstehenden Anwendungsoberflächen vereinen die klassischen Merkmale einer Windows-GUI mit den Möglichkeiten eines browsergestützten Internets, denn letztlich handelt es sich bei jedem Silverlight-Programm um eine Weboberfläche, auch wenn das dem Benutzer nicht mehr bewusst ist. Der Vorteil liegt hier ganz klar darin, dass man die neuesten Versionen des Gesamtpakets nicht mehr im ganzen Haus verteilen muss, sondern eine zentrale Stelle, einen Server, hat, auf dem die Anwendung installiert ist. Die Mitarbeiter starten nur noch ihren Browser, klicken auf den entsprechenden Link und haben sofort ihre Lohnbuchhaltung oder Lagerverwaltung vor sich. Über die OutOfBrowser-Funktionalität von Silverlight ist ein Anwender zudem in der Lage, seine wichtigsten Programme auf seinem Desktop zu installieren, womit auch noch der letzte Hinweis verloren geht, dass er mit einer Browseroberfläche arbeitet.

Damit sind diese Anwendungen auch von außen erreichbar, wenn man dies will. Das Einzige, was der Vertriebsbeauftragte im Außendienst benötigt, ist eine UMTS-Verbindung und schon kann er die Auftragspositionen direkt im System erfassen und mit dem Kunden über realistische Liefertermine verhandeln.

Wie bereits beschrieben, werden mit Silverlight auch die Ersteller eher kreativer Oberflächen angesprochen. Diese müssen immer online sein, weil sie neue Kun-

den erreichen und den Internetauftritt des Unternehmens qualitativ aufwerten wollen. Immerhin ist das die wichtigste Visitenkarte, die man als Unternehmer hat. Eine billige, selbst gestrickte, starre HTML-Seite mit einem total langweiligen Kontaktformular als einzige Interaktionsmöglichkeit schreckt viele ab. Jedes Unternehmen, das sich aktiv präsentieren muss, also im Vertrieb tätig ist, braucht hier eine perfekte Lösung – die Bäckerei um die Ecke gehört da allerdings nicht dazu.

In der Einleitung wurde bereits darauf eingegangen, dass dieses Buch im Prinzip zweigeteilt ist. Grob gesagt, beschäftigt sich der eine Teil mit den Grafikgrundlagen, mit Animationen und Videos, während der andere immer wieder auf Controls eingeht, die sich natürlich auch etwas effektvoller gestalten lassen. Am Ende wird dann eine einfache, aber doch komplette Anwendung zur Fehlerverfolgung gezeigt, die sich sowohl um Benutzerführung, Dialogsteuerung, Anwendungslogik als auch um Datenbankzugriffe kümmert.

## 1.2 Silverlight und Flash

Wenn man sich dem Thema Silverlight annähert, wird man irgendwann ganz automatisch mit der Frage konfrontiert, warum man denn nicht Flash nehmen sollte. Immerhin gibt es diese Technologie schon seit über zehn Jahren und sie ist stark verbreitet. Das ist durchaus richtig und wenn man eher ein Werbebeauftragter ist, dürfte es auch keine große Rolle spielen, ob man Silverlight oder Flash benutzt. Man wird aber in Zukunft mit Sicherheit beides können müssen, weil die Marktmacht von Microsoft zwangsläufig dazu führt, dass Silverlight einen mindestens beachtlichen Verbreitungsgrad bekommen wird. Wenn man im Internet nach den Begriffen »Silverlight« und »Flash« sucht, bekommt man viele Seiten angeboten, die hier einen Vergleich ziehen wollen. Da man jedoch nur in den seltensten Fällen den jeweiligen Autor und seine persönliche Ausrichtung kennt, sind diese Vergleiche stets mit Vorsicht zu genießen, denn es ist natürlich immer gerade das am besten, was man selbst beherrscht. Dennoch ist auf einer dieser Seiten folgender Abschnitt zu lesen, der die Unterschiede zwischen beiden Ansätzen ziemlich genau wiedergibt.

*Vergleicht man beide Ansätze, so fällt Folgendes auf: Flash entwickelte sich vom Illustrations- und Animationstool zur ernsthaften Programmiersprache im Bereich der Webentwicklung. Silverlight geht scheinbar den Weg umgekehrt und bringt in eine Umgebung, die eigentlich bisher nur Anwendungsentwicklern vorbehalten war, plötzlich Animation und Grafik mit ein.*

Es ist die tiefe Integrationsmöglichkeit von Silverlight in die relevanten Business-Anwendungen im Haus, die diesen Ansatz so interessant macht. Sämtliche Entwicklungsumgebungen wie Visual Studio oder der Visual Web Developer sind den Anwendungsentwicklern bekannt und die notwendige Logik hinter der Fassade

wird mit Visual Basic und/oder C# programmiert. Da aber gute Entwickler oft miserable Grafiker sind, wird der eigentliche Auftritt, also das letztendliche Aussehen der Oberfläche, außerhalb des Quellcodes mit Hilfe von XML beschrieben, was durch Tools wie Expression Blend hervorragend unterstützt ist. Beide Teams, Grafiker und Entwickler, arbeiten mit unterschiedlichen Werkzeugen gleichzeitig am selben Projekt und haben so sofort die notwendige Rückkopplung.

## 1.3 WPF und XAML

Der Begriff XAML spielt in Silverlight eine zentrale Rolle und steht ausgeschrieben für eXtensible Application Markup Language. Es handelt sich also um eine nach XML aufgebaute Sprache zur Beschreibung von Anwendungen.

Der andere wichtige Begriff ist WPF und steht für Windows Presentation Foundation, ein Framework zur Ausgabe von Grafiken, Videos, Audio und Bildern, das auf DirectX aufsetzt.

Silverlight basiert auf beidem und bietet über XAML einen Ausschnitt aus WPF für die Programmierung an. Einem erfahrenen WPF-Entwickler kann es passieren, dass er in Silverlight nicht all das vorfindet, was er gewohnt ist. Silverlight soll möglichst leichtgewichtig sein und deswegen sind nicht alle WPF-User Controls oder nur in abgespeckter Form verfügbar.

```
001 <UserControl x:Class="EinButton.MainPage"
002     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
003     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
004     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
005     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/
006         2006"
007     mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480">
008     <Canvas HorizontalAlignment="Center" VerticalAlignment="Center">
009         <Rectangle Height="100" Width="150" Fill="Orange"
010             Stroke="Black"
011             x:Name="Rechteck"
012             MouseLeftButtonDown="Rechteck_MouseLeftButtonDown"/>
013         <TextBlock Canvas.Left="35" Canvas.Top="40"
014             FontFamily="Comic Sans MS">
015             Bitte anklicken
016         </TextBlock>
017     </Canvas>
018 </UserControl>
```

Listing 1.1: Eine einfache XAML-Datei

Um einen ersten Eindruck zu geben, ist in Listing 1.1 eine ganz einfache XAML-Datei abgedruckt, die ein orangefarbiges Rechteck zeichnet, das in seiner Mitte mit dem Text *Bitte anklicken* beschriftet ist. Das Beispiel stammt aus Kapitel 3, wo es ganz ausführlich erklärt wird.

Der Anwendungsentwickler erzeugt den Inhalt dieser Datei nur so weit, wie er es für seine Arbeit benötigt. In Zeile 10 lenkt er beispielsweise den Event, dass der Benutzer mit der linken Maustaste auf das Rechteck klickt, in eine Methode mit dem Namen `Rechteck_MouseLeftButtonDown` um, die er beispielsweise in C# programmiert hat. Um das testen zu können, braucht er eine funktionierende Oberfläche – wie die aussieht, ist erst mal egal.

Wenn der Designer diese XAML-Datei aufnimmt, dann interessieren ihn vorrangig die Proportionen, die verwendeten Farben, hübsche Schatten und eine ansprechende Schriftart.

Genau aus diesem Grund, weil Entwickler und Designer unterschiedliche Sichtweisen haben, wurde die Trennung zwischen Logik und Präsentation vollzogen. Jeder muss vom anderen gerade so viel wissen, wie unbedingt notwendig ist.

## 1.4 CodeBehind

Der CodeBehind, von dem immer wieder die Rede sein wird, ist das Stück Quellcode, das abgearbeitet wird, wenn auf der Oberfläche irgendetwas getan wird. Wenn also der Benutzer mit der linken Maustaste auf das Rechteck klickt, dann will er dafür belohnt werden. Es muss irgendetwas passieren, am besten noch das, was sich der Anwender davon erwartet.

In welcher Sprache der CodeBehind geschrieben wird, ist eigentlich egal. Im Rahmen dieses Buches wird ausschließlich C# benutzt. In einem der letzten Kapitel wird gezeigt, wie man von einer Silverlight-Anwendung aus, die den Client einer Business Application repräsentiert, mit Hilfe von WebServices mit einer serverseitigen Implementierung kommuniziert, deren Programme das Haus nie verlassen werden. Sobald man über WebServices geht, kann der Teil, den man hier anspricht, auch in Java geschrieben sein. Bei dem Zielsystem kann es sich z.B. um ein applikationsservergestütztes JavaEE-SessionBean handeln.

```
...
013 namespace EinButton
014 {
015     public partial class MainPage : UserControl
016     {
    ...
022         public MainPage()
023         {
```

```
024     InitializeComponent();
025     }
026
027     private void Rechteck_MouseLeftButtonDown(object sender,
028         MouseButtonEventArgs e)
029     {
030         ...
038     }
039 }
040 }
```

**Listing 1.2:** Beispielhafter CodeBehind

In Listing 1.2 findet sich der passende Codeschnipsel für die bereits erläuterte Methode `Rechteck_MouseLeftButtonDown`. Die drei Punkte innerhalb der Zeilennummern deuten die weggelassenen Stellen an. Auch hier der Hinweis, dass diese Klasse komplett in Kapitel 3 beschrieben ist.

Was in dieser Methode geschieht, ist an dieser Stelle unwichtig. Es soll gezeigt werden, wo man die Oberfläche definiert und wo man ihre Logik programmiert. Selbstredend kann diese Methode auch innerhalb einer Debugsession abgearbeitet werden, um so herauszufinden, warum das Programm nicht so will wie man selbst.

## 1.5 Unterstützte Betriebssysteme

Silverlight ist von Microsoft und deshalb liegt natürlich sofort die Vermutung nahe, dass die geschriebenen Anwendungen nur unter Windows und so richtig nur mit dem Internet Explorer lauffähig sind.

Weit gefehlt!

Neben Windows wird ausdrücklich das Betriebssystem Mac OS X 10.4.8 unterstützt und auch für Linux, FreeBSD und SolarisOS stehen die Silverlight-Anwendungen in Firefox zur Verfügung.

Tatsächlich ist Microsoft bestrebt, mit Silverlight ein plattformunabhängiges Werkzeug zu schaffen, weil es nur dann die notwendige Akzeptanz finden wird. Es ist erklärtes Ziel, sich hier nicht nur auf Windows zu konzentrieren.

## 1.6 Unterstützte Browser

Auch die Liste der unterstützten Browser geht über den Internet Explorer hinaus. Explizit genannt sind Firefox, der für alle Beispiele im Rahmen dieses Buches verwendet wurde, und Safari.

## 1.7 Unterstützung für Windows Phone 7

Wer Anwendungen für das Windows Phone 7 schreiben will, ist bei Silverlight genau richtig. Aktuell sind der Silverlight 3 Standard und einige zusätzliche Features auf dem Windows Phone verfügbar. Damit benötigt man lediglich eine einzige Entwicklungsumgebung, um sowohl Inhouse-Anwendungen, Internetauftritte und Programme für mobile Endgeräte erstellen zu können. In der Dokumentation zur Silverlight Klassenbibliothek von Microsoft findet sich stets auch ein Hinweis, ob die einzelne Klasse und Methode auch auf dem Windows Phone genutzt werden kann.

## 1.8 Beispiele aus diesem Buch

Alle Beispiele, die im Rahmen dieses Buches erstellt und erklärt wurden, können kostenlos aus dem Internet unter der Adresse [www.miptp.de/9074](http://www.miptp.de/9074) bezogen werden.

Wenn man die Datei entpackt, entsteht ein Verzeichnis mit dem Namen `Silverlight4` mit einem weiteren Unterverzeichnis `Projekte`. Letzteres enthält dann pro Kapitel die einzelnen Beispiele in Form fertiger Projektordner.

Die gesamte Entwicklungsplattform für Silverlight kann kostenlos aus dem Internet bezogen werden. Lediglich das Programm Expression Blend gibt es nur in einer 60-Tage-Testversion. Im nächsten Kapitel ist ausführlich erklärt, wo man sich diese Programme kopieren kann und wie man sie installieren muss. Dabei spielt es keine Rolle, ob man sich für die deutsche oder englische Variante entscheidet. Wenn im Rahmen dieses Buches Bezug auf einen Menüeintrag genommen wird, dann ist dieser immer in beiden Sprachen abgedruckt.

## 1.9 Next Steps

Im nächsten Kapitel wird gezeigt, wie man sich die benötigte Entwicklungsumgebung aufbaut. Sie sollten die Beispiele wirklich durcharbeiten, weil das immer noch die effizienteste Möglichkeit ist, die Programmierung von Silverlight zu lernen. Daher sollten Sie sich auch die Zeit nehmen, die einzelnen Tools wie dort beschrieben zu installieren. Hier bietet es sich natürlich an, dies innerhalb einer eigenen, virtuellen Maschine zu machen, wie Sie sie sich beispielsweise mit dem ebenfalls kostenlosen Programm Virtual PC von Microsoft erstellen können.

In Kapitel 3 wird dann ein ganz einfaches Beispiel erstellt und sehr genau erklärt, wie ein Silverlight-Projekt aufgebaut ist. Hier erfahren Sie auch, wie man den Debugger einsetzt, um möglichst schnell eigene Fehler finden und beseitigen zu können. Für erfahrene Entwickler ist das eine Selbstverständlichkeit.

Gleich danach geht es zu den grafischen Grundlagen von Silverlight, bevor dann immer mehr Gewicht auf die eigentliche Anwendungsentwicklung gelegt wird.