

Silverlight

Wechseln zu: [Navigation](#), [Suche](#)

Inhaltsverzeichnis

- [1 Definition](#)
- [2 Architektur von Silverlight](#)
 - [2.1 Core-Presentation-Framework](#)
 - [2.2 NET-Framework for Silverlight](#)
- [3 Applikations- und Programmiermodell](#)
- [4 Systemanforderungen](#)
- [5 Quellen](#)

1 Definition

Microsoft veröffentlicht mit Silverlight eine browser- und plattformunabhängige Implementierung des NET-Frameworks vornehmlich zum Erstellen von reichhaltigen Internet Applikationen. Von verwaltetem Code und dynamischen Sprachen über traditionelle und deklarative Programmierung kann hier auf die meiste Funktionalität des NET-Frameworks zugegriffen werden. Mit der ebenfalls neu veröffentlichten Windows Presentation Foundation (WPF) werden nicht nur dem Entwickler sondern auch dem Designer in seiner Rolle jede Menge Möglichkeiten geboten, hoch ansprechende User Interfaces zu entwickeln bzw. zu designen.

2 Architektur von Silverlight

Silverlight bietet nicht nur das Darstellen von medialen Inhalten, wie Bildern, Video und Audio im Browser, vielmehr ist es darüber hinaus eine leichtgewichtige Plattform zur Entwicklung von verteilten Anwendungen, welche diverse Datenquellen integrieren und Servicequellen konsumieren. Dabei steht dem Entwickler ein Subset an Funktionalität des umfangreichen, mit derzeit sage und schreibe ca. fünfundzwanzigtausend Klassen starken NET-Library zur Verfügung. Die Silverlight-Plattform an sich besteht dabei aus zwei Kernkomponenten einschließlich einer Installations- und Update-Komponente:

2.1 Core-Presentation-Framework

Das CorePresentationFramework stellt Komponenten und Services rund um das User Interface (UI) bereit und gewährleistet die Interaktion mit dem Benutzer. Darin enthalten sind neben leichtgewichtigen Web-Steuerelementen, Module zum Abspielen von Videos, DataBinding und präsentationsorientierte Funktionalität wie beispielsweise Vektorgraphiken und Textanimationen. Darüber hinaus beinhaltet diese Komponente die Extensible Application Markup Language (XAML), welche als deklarative Programmiersprache in einer XML-ähnlichen Notation als primäres Interaktionsmedium zwischen dem .Net-Framework und der Präsentationsschicht agiert. XAML gewährleistet Entwicklern und Designern die Spezifikation des Layouts und weiterer, die

Präsentationsschicht betreffender Funktionalität. Neben diesem Ansatz kann eine Manipulation der Präsentationsschicht auch programmatisch mit Managed-Code (C# oder VB.NET) erfolgen.

2.2 NET-Framework for Silverlight

Das .NetFramework for Silverlight als zweite große Komponente der Silverlight-Plattform erweitert diese um auserlesene Komponenten und Bibliotheken des .NetFrameworks. Konkret werden in diesem Umfang die BaseClassLibraries mit grundlegenden Funktionalitäten zur Bearbeitung von Zeichenketten, Ein-/Ausgabe-Funktionalität, Reflection, RegularExpressions, Collections und Globalization integriert. Einen weiteren essentiellen Bestandteil stellt die CommonLanguageRuntime (CLR) dar. Diese bietet neben der Speicherverwaltung und der automatischen Speicherbereinigung (Garbage collection) Mechanismen zum Überprüfen der Typsicherheit und der Ausnahmebehandlung. Der Datenzugriff kann in Silverlight mit der neu eingeführten Abfragesprache LINQ (Language Integrated Query) umgesetzt werden. Kommunikation von Anwendung über lokale Rechengrenzen hinweg wird durch die ebenfalls neue Windows Communication Foundation (WCF) ermöglicht, welche in der abgespeckten Version Unterstützung für cross-domain HTTP Requests, RSS/ATOM, JSON, POX und SOAP bietet. Neben dem Verwenden von Managed-Code, kann mit Hilfe der DLR (Dynamic Language Runtime) auch auf dynamische Sprachen wie IronPython und IronRuby zurückgegriffen werden.

3 Applikations- und Programmiermodell

Seit Microsoft am 14. Oktober 2008 die finale Version von Silverlight 2.0 der Öffentlichkeit vorgestellt hat, besteht für Entwickler endgültig die Notwendigkeit, Silverlight bei der Planung einer Webanwendung als Alternative, etwa zu ASP.NET oder einer AJAX-gestützten Lösung, in Betracht zu ziehen. Dabei spielen neben dem eigenen Vermögen, eine Anwendung auf Basis von Silverlight erstellen zu können, weitere wichtige Faktoren eine Rolle, wie bspw. Fragen in Bezug auf die Infrastruktur. Welchen Server setzt der Kunde ein? Welche Tools und Entwicklungsumgebungen stehen zur Verfügung und darüber hinaus und zu guter Letzt, welche Fähigkeiten besitzen die Entwickler? Die meisten, zukünftig mit Silverlight realisierten Anwendungen werden wohl auf der Version 2.0 aufsetzen und die Funktionalität der vorkompilierten Klassen der Common Language Runtime (CLR) oder der Dynamic Language Runtime (DLR) nutzen. Dennoch ist es notwendig zu wissen, dass mit Silverlight 1.1 ein völlig anderer Ansatz und ein anderes Applikations- bzw. Programmiermodell gewählt werden kann. Diese Version verwendet dabei anstelle von verwaltetem Code JavaScript-Code, welcher durch den Browser auf Clientseite interpretiert wird. Die Wahl der zu verwendenden Version sollte vor Projektstart gut durchdacht sein, eine parallele Verwendung der beiden Versionen wird nicht unterstützt.

4 Systemanforderungen

[Visual Studio 2008 Service Pack 1](#)

[Silverlight-Tools für Visual Studio 2008](#)

[Expression Blend 2 Service Pack 1](#)

5 Quellen

Die offizielle Silverlight-Webseite
Silverlight in der MSDN

Kategorien:

[Web-Programmierung](#)

[Content-Management](#)

[World Wide Web](#)

[Glossar](#)

Diese Seite wurde zuletzt am 30. Januar 2009 um 20:29 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

