

.NET + Silverlight

BOUW SILVERLIGHT-APPLICATIES IN .NET

Een belangrijk onderdeel van Microsoft's strategie voor de volgende generatie interactieve websites is Silverlight, waarvan deze zomer versie 1.0 is uitgekomen. Ondertussen is Microsoft echter al druk bezig met versie 1.1 van Silverlight. Deze heeft als voornaamste speerpunt een ingebouwde .NET-runtime die werkt met Safari, Firefox en IE op Mac en Windows. Kortom: nieuwe mogelijkheden voor developers en designers.

In dit artikel kijken we naar de mogelijkheden van Silverlight 1.1 alpha en zien we hoe een XAML-gebruikersinterface wordt gekoppeld aan client-side .NET-code. Als voorbeeld toont dit artikel hoe de Silverlight XAML-klok op drie verschillende manieren kan worden geprogrammeerd: met JavaScript, met Script# (de C# naar JavaScript-compiler) of direct met C# in Silverlight 1.1 .NET.

Silverlight = WPF voor het web?

Samen met ASPNET Ajax is Silverlight het platform van Microsoft voor het bouwen van Rich Internet Applications. De belangrijkste karakteristieken van een RIA zijn:

- Automatische uitrol zonder installatie
- Rijke en interactieve gebruikersinterface
- Cross-browser en cross-platform

Silverlight voldoet aan al deze RIA-voorwaarden. Technisch gezien is Silverlight een browser plug-in die de in XAML (eXtensible Application Markup Language) gedefinieerde UI-elementen kan tonen. Silverlight werkt cross-browser (IE, Firefox en Safari) en is beschikbaar voor Windows en Mac. Microsoft werkt momenteel aan een Silverlight-versie voor Windows Mobile en door Mono-ontwikkelaars wordt zelfs al gewerkt aan een versie voor Linux met codenaam 'Moonlight'. Verder heeft Silverlight een eigen object- en eventmodel dat geschikt is voor scripting in de browser. Oftewel: een grafisch ontwerper kan een XAML-document maken met plaatjes, vectoren en animaties, die een ontwikkelaar vervolgens kan voorzien van

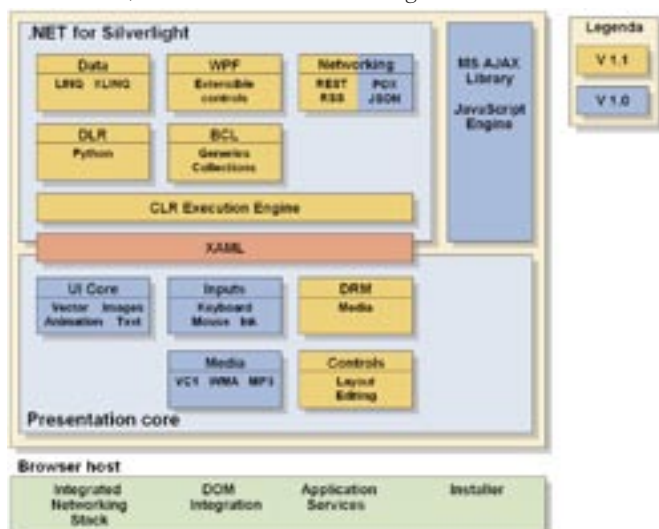
code om de gewenste functionaliteit te krijgen. Silverlight is gebaseerd op de WPF UI-elementen in .NET Framework 3.0, maar bevat daarvan slechts een deelverzameling; zo worden onder andere databinding, styling, controls en 3D niet standaard ondersteund in Silverlight 1.0. Maar in tegenstelling tot WPF bevat Silverlight wel weer mogelijkheden van streaming audio/video en interactie met html-elementen op de webpagina. Het beste kun je Silverlight dan ook vergelijken met Flash. Als je Silverlight gaat vergelijken met Flash is natuurlijk de eerste vraag waarom het ene beter zou zijn dan het andere. Flash is – omdat het langer in de markt is – momenteel op meer dan 95% van alle computers geïnstalleerd; iets dat Microsoft met Silverlight nog niet heeft bereikt. Maar Silverlight ondersteunt als enige de VC-1-codec voor 'high definition' videocontent en het percentage pc's waarop Silverlight is geïnstalleerd, zal ongetwijfeld heel snel stijgen. Op het eerste gezicht lijken de features van Silverlight en Flash veel op elkaar. Maar met de aankomende 1.1-release van Silverlight heeft Microsoft een troef in handen waarvan iedere .NET-softwareontwikkelaar het grote voordeel direct ziet: integratie met .NET en Visual Studio.

Silverlight 1.1: programmeren in .NET

Silverlight 1.1 bevat alle features van Silverlight 1.0 en heeft daarnaast een ingebouwde (maar afgeslankte) versie van de .NET-runtime. Je hoeft dus niet het .NET Framework geïnstalleerd te hebben om het te gebruiken, want Silverlight 1.1 levert zelf een cross-platform .NET-versie mee. Dat betekent dat je Silverlight-applicaties in managed .NET-code kunt programmeren, in tegenstelling tot Silverlight 1.0 dat alleen JavaScript ondersteunt. Hierdoor ben je als ontwikkelaar veel productiever in het bouwen van Silverlight RIA-applicaties, omdat de programmeertaal (VB.NET, C#, et cetera) en omgeving (Visual Studio) bekend en vertrouwd zijn. Je kunt zelfs de Silverlight-applicaties client-side debuggen in Visual Studio. Voor grafische vormgevers is er bovendien Expression Blend beschikbaar, waarmee je een design eenvoudig kunt vertalen naar Silverlight XAML-elementen.

De .NET CLR in Silverlight 1.1 is gebaseerd op het .NET Framework 2.0, maar is daar wel een subset van. Om een indruk te geven: alle .NET Framework 2.0 DLL's bij elkaar hebben een omvang van 50MB, terwijl de Silverlight .NET DLL's maar 4MB in beslag nemen. Ook heeft de .NET-runtime van Silverlight 1.1 beperkte rechten, omdat het in de browser draait. Dit wordt ook wel 'sandboxing' genoemd. Eventjes iets van het filesysteem of uit de registry lezen gaat dus niet. Silverlight 1.1 biedt daarentegen wel weer een afgeschermd opslagmogelijkheid van tijdelijke data ('isolated storage').

Opsommingen zijn vaak saai, maar de features van Silverlight 1.1



Afbeelding 1. De Silverlight-systeemclasses

```

<html>
<head>
  <title>Silverlight 1.1 voorbeeld</title>
  <script type="text/javascript" src="Silverlight.js"></script>
</head>
<body>
  <div id="ControlHost">
    <script type="text/javascript">
      Sys.Silverlight.createObjectEx({
        source: "xaml/MijnApp.xaml",
        parentElement: document.getElementById("ControlHost"),
        id: "MijnApp",
        properties: {
          width: "310",
          height: "310",
          version: "0.95", //0.95 = Silverlight 1.1 alpha
          background: "#00000000",
          isWindowless: true,
          enableHtmlAccess: true
        },
        events: {}
      });
    </script>
  </div>
</body>
</html>

```

Codevoorbeeld 1. MijnApp.htm-hostpagina voor een Silverlight-control

- wil ik jullie toch niet onthouden; zie ook afbeelding 1:
- Bevat alle mogelijkheden van Silverlight 1.0
 - Een cross-platform .NET-runtime (CLR)
 - Programmeer Silverlight in de .NET-taal van jouw keuze
 - 1MB 'isolated disk-storage' ter beschikking van jouw applicatie
 - Visual Studio-integratie met onder andere debugging, refactoring en IntelliSense
 - Ondersteuning van Dynamic Languages (DLR) zoals IronPython en JScript
 - Standaard controls zoals Button, Scrollbar, Slider en Listbox
 - Uitbreidbaarheid van XAML en controls
 - Web Service client-proxy generatie (alleen in VS Orcas)
 - LINQ (Language Integrated Query) en generics ondersteuning
 - Interactie tussen HTML en managed .NET-code
 - Http-networking, XmlReader en XmlWriter.

Silverlight 1.1 is nu al beschikbaar als alfa-versie, en werkt met Visual Studio 2005 en Visual Studio 'Orcas'-beta-versie 1. Voor VS Orcas is bovendien de 'Silverlight tools'-alpha beschikbaar, waarmee Silverlight echt een first-class citizen wordt in Visual Studio. Als je direct van start wilt gaan, kijk dan op <http://silverlight.net/GetStarted>.

```

<Canvas x:Name="mijnApp"
  Opacity="0"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="MijnApp;assembly=clientbin/MijnApp.dll">

  <!-- Voorbeeld TextBlock control -->
  <TextBlock x:Name="MijnLabel" MouseEnter="MijnLabel_MouseEnter"/>

  <!-- Voorbeeld XAML figuur -->
  <Path Data="M 157, 5 a 150,150 0 1,0 1,0 z">
    <Path.Fill>
      <SolidColorBrush Color="Black" Opacity="0.3"/>
    </Path.Fill>
  </Path>
  ...
</Canvas>

```

Codevoorbeeld 2. MijnApp.xaml-canvas met codebehind x:Class

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;

public partial class MijnApp : Canvas
{
  public void MijnLabel_MouseEnter(object sender, MouseEventArgs e)
  {
    ((TextBlock)sender).Text = "Muis over label";
  }
}

```

Codevoorbeeld 3. Implementatie van een eventhandler in Silverlight 1.1

Silverlight 1.1: de basisstructuur

De basisstructuur van een Silverlight 1.1-applicatie is eigenlijk vrij eenvoudig. In het kort komt het er op neer dat vanuit een host-pagina (dit mag een gewone html-pagina zijn) via script het Silverlight-control op de pagina wordt aangemaakt. Indien Silverlight nog niet is geïnstalleerd op de pc van de bezoeker, dan zal het script dit opmerken en een 'install now'-knopje tonen. Nadat het Silverlight-control is aangemaakt, zal het een aangegeven xaml-file inlezen en tonen. In Silverlight 1.1 kun je in deze xaml ook een (client-side!) code-behind DLL aangeven, die vervolgens door de browser vanaf de server wordt opgehaald en in het geheugen van het betreffende Silverlight-control wordt geladen.

Een .NET Silverlight 1.1-applicatie bestaat altijd minimaal uit:

- Een html- of ASPX-pagina om een Silverlight-control in te 'hosten' - MijnApp.htm
- Het Silverlight-load script - Silverlight.js
- Een root XAML- & .NET-assembly - MijnApp.xaml & MijnApp.dll

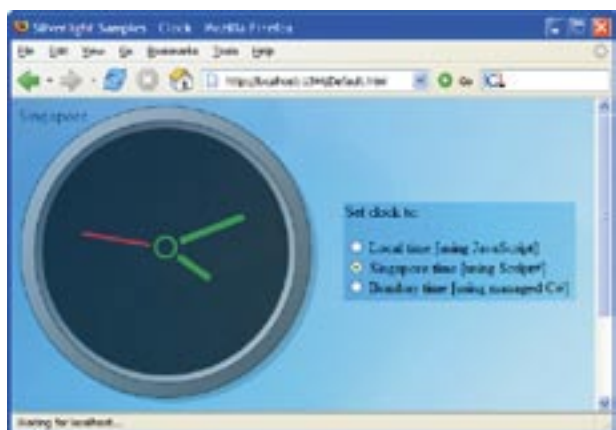
In codevoorbeeld 1 is de inhoud van de Silverlight-host-pagina 'MijnApp.htm' te zien, waar het Silverlight-control wordt gecreëerd en gekoppeld aan 'MijnApp.xaml'. In codevoorbeeld 2 zien we een gedeelte van de inhoud van 'MijnApp.xaml' met daarin een referentie naar de client-side .NET-assembly 'MijnApp.dll'.

XAML koppelen aan .NET code-behind

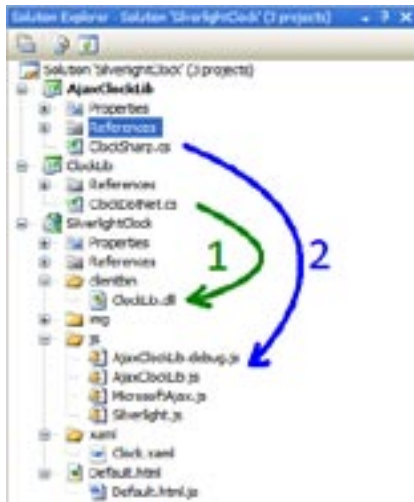
Omdat 'MijnApp' als .NET-class is opgegeven in de XAML (zie x:Class-attribuut in codevoorbeeld 2), kunnen we code voor gebruikersinteractie implementeren op een manier zoals we nu al gewend zijn in ASP.NET of WinForms. In codevoorbeeld 3 zie je de inhoud van 'MijnApp.cs', waarin een eventhandler gedefinieerd staat voor het MouseEnter-event op het XAML TextBlock met naam 'MijnLabel'.

De Silverlight XAML-klok programmeren

De voorbeeldcode uit dit artikel is te downloaden van www.microsoft.com/net/magazine/18 en laat drie verschillende manieren zien om de Silverlight XAML-klok via code in te stellen op een gewenste tijd (zie afbeelding 2), namelijk:



Afbeelding 2. Screenshot van de XAML-klokdemo



Afbeelding 3. De Silverlight-klok voorbeeld-solution

- Met JavaScript (Silverlight 1.0 en 1.1)
- Met Script# en Ajax (Silverlight 1.0 en 1.1)
- Met C# in Silverlight 1.1 .NET

Omdat het zelf schrijven van cross-browser JavaScript erg lastig en foutgevoelig is, laat ik in de voorbeeldcode ook een alternatief hiervoor zien: Script# van ASP.NET-architect Nikhil Kothari. Script# is een code-generatietool die C# vertaalt naar cross-browser JavaScript. Script# heeft een eigen (beperkte) base class-library. Als je een Script#-project compileert in Visual Studio, dan is het resultaat geen IL-assembly, maar JavaScript dat op de browser kan worden geïnterpreteerd. Script# is oorspronkelijk bedoeld om JavaScript te kunnen genereren voor interactie met elementen op de html-pagina, maar sinds eind 2006 ondersteunt Script# ook het objectmodel van Silverlight 1.0.

Uiteindelijk is Silverlight 1.1 met de ingebouwde .NET-runtime van de drie oplossingen de krachtigste, vooral vanwege de rijke base class-library en de Visual Studio-integratie met onder andere client-side debugging. Indien je geïnteresseerd bent in Silverlight- development, dan nodig ik je uit om de voorbeeldcode te downloaden en er zelf mee te experimenteren. Afbeelding 3 toont ter verduidelijking dat het Silverlight .NET-project 'ClockLib' compileert naar 'clientbin/ClockLib.dll' (groene pijl 1) en dat het Ajax/Script#-project 'Ajax-ClockLib' compileert naar 'js/AjaxClockLib.js' (blauwe pijl 2).

Om de voorbeeldcode uit te proberen, dien je het volgende geïnstalleerd te hebben:

- Visual Studio 2005 Standaard (of hoger)
- Silverlight 1.1 alpha (<http://silverlight.net/GetStarted>)
- Microsoft Ajax library (<http://ajax.asp.net/downloads>)
- Script# 0.3 (<http://projects.nikhilk.net/Projects/ScriptSharp.aspx>)

Krachtig platform

De aankomende versie 1.1 van Silverlight biedt vanwege zijn ingebouwde .NET Framework en de integratie met Visual Studio een erg krachtig platform voor het bouwen van RIA-browserapplicaties. Houd dit jaar de aankomende bètaversies van Silverlight 1.1 goed in de gaten, want er zullen steeds meer mogelijkheden aan dit platform worden toegevoegd.

Dion Olsthoorn is softwarearchitect bij Macaw (www.macaw.nl) en werkzaam bij de businessunit 'Rich Internet Solutions'. Naast zijn werkzaamheden als architect is het schrijven van code en het onderzoeken van nieuwe technologieën nog altijd zijn grote passie. Dion is te bereiken op dion.olsthoorn@macaw.nl en zijn weblog staat op <http://weblogs.asp.net/dmetjes>.

Referenties

Voorbeeldcode bij dit artikel: <http://www.microsoft.com/net/magazine18>

Microsoft's Silverlight-site: <http://www.silverlight.net>

Silverlight 'Get Started': <http://silverlight.net/GetStarted>

Script# 0.3: <http://projects.nikhilk.net/Projects/ScriptSharp.aspx>

Silverlight 1.1 'Developer Reverence'-poster: <http://tinyurl.com/2f82c5>

