

Silverlight 2 introductie

SILVERLIGHT MET EEN .NET-SMAAKJE

Vorig jaar stal Silverlight 1.0 de show op Mix in Las Vegas. Dit jaar maakte versie 2 van het product zijn opwachting. Developers kunnen Silverlight-applicaties bouwen met volledige ondersteuning van het .NET Framework en moeten dus niet langer Javascript-code schrijven om RIA (Rich Internet Applications) te ontwikkelen met het nieuwe platform.

Silverlight versie 2 biedt heel wat nieuwe mogelijkheden aan developers, meer zelfs: het product bevat zoveel interessante features dat het onmogelijk is deze allemaal in één artikel te behandelen. Bedoeling is dan ook een beknopt overzicht te geven van de meest opvallende nieuwigheden en de belangrijkste verschillen tussen V1.0 en V2. Tenslotte gieten we dit alles samen in een demo waarin we met Silverlight 2 een eenvoudige RSS-feed reader bouwen.

1.1 werd 2

Het web van vandaag vraagt om interactieve pagina's. Het model achter het web blijft echter steeds hetzelfde: een browser ontvangt content van een server en toont deze. Met Javascript krijgen developers toegang tot de elementen binnen het Document Object Model. Door het manipuleren van deze elementen kan interactiviteit worden gecreëerd. Echter, wat met Javascript kan bereikt worden, is enigszins beperkt: de user-experience (UX) die vandaag verwacht wordt, is niet meer alleen (of toch zeer moeilijk) te bereiken met alleen Javascript. Microsoft heeft in 2007 Silverlight 1.0 uitgebracht: een browser plug-in die het maken van RIAs sterk vereenvoudigt. V1.0 bestond voornamelijk uit een XAML-rendering engine en 'script-behind'. Developers moesten Javascript-code schrijven om deze XAML-elementen aan te sturen. Deze versie was dan ook vooral gericht op media en graphics. Naast V1.0 werd ook V1.1 aangekondigd als alpha-release, waarover je in een vorig nummer van dit magazine al kon lezen. Om alle verwarring te voorkomen: Silverlight 1.1 werd eind 2007 omgedoopt tot Sil-

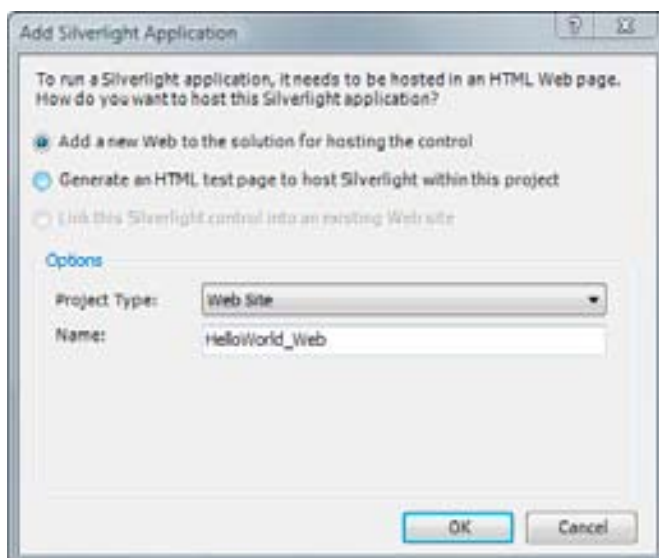
verlight 2. De toevoeging van onder andere de Common Language Runtime (CLR) en Base Class Libraries (BCL,) maakten duidelijk dat het om een grote release ging.

Hello Silverlight 2

Silverlight 2 is net zoals zijn voorganger een cross-browser plug-in, beschikbaar voor de meest gangbare browsers (IE, FireFox en Opera) op Windows en Mac (in de toekomst ook op Linux onder de naam MoonLight). De belangrijkste toevoegingen zijn de .NET CLR (Common Language Runtime) en een subset van de .NET Base Class Library. In deze subset zitten onder meer LINQ-To-Objects, generics en Isolated Storage. Developers kunnen met versie 2 applicaties schrijven met Visual Basic, C#, Managed Jscript en dynamische talen zoals IronRuby. Net als de plug-in, is de afgeslankte versie van het .NET Framework cross-platform te gebruiken. De XAML-code is overigens ook een subset van WPF, maar bevat ook een aantal controls, zoals de DataGrid, die nog niet in WPF aanwezig zijn. Op deze manier wordt het schrijven van eenzelfde applicatie voor zowel de desktop met WPF als voor de browser met Silverlight veel eenvoudiger. De plug-in, inclusief de CLR, is zo'n 4MB groot. Voor alle duidelijkheid: versie 2 vereist geen aanwezigheid van de volledige versie van het .NET Framework. Voor deployment kan overigens elk type server gebruikt worden, inclusief Apache. Wel moet Silverlight binnen een pagina gehost worden. Deze pagina kan zowel ASPNET, PHP als HTML zijn. Op het moment van schrijven, is Silverlight 2 bèta 1 met een niet-commerciële go-live-licentie beschikbaar. Microsoft heeft aangekondigd om bèta 2 uit te brengen in het tweede kwartaal van dit jaar. Release 2 is overigens compatibel met de vorige versie en alle applicaties die zijn geschreven voor de vorige versie, werken ook in de nieuwe.

Programmeren in 2

Naast de 2 Beta 1 run-time, heeft Microsoft ook de 'Silverlight 2 tools for Visual Studio 2008 and SDK' beschikbaar gesteld. Hiermee kunnen ontwikkelaars aan de slag om Silverlight



Afbeelding 1. Voeg een testwebsite toe

```
<UserControl x:Class="HelloWorld.Page" Width="400" Height="300">
  <Grid x:Name="LayoutRoot" Background="White">
    <TextBlock Text="Hello World"></TextBlock>
  </Grid>
</UserControl>
```

Codevoorbeeld 1.

```
namespace HelloWorld {
  public partial class Page : UserControl {
    ...
  }
}
```

Codevoorbeeld 2.

```
<object data="data:application/x-silverlight,"
type="application/x-silverlight-2-bl" >
  <param name="source" value="ClientBin/HelloWorld.xap"/>
  <param name="onerror" value="onSilverlightError" />
  <param name="background" value="white" />
  ...
</object>
```

Codevoorbeeld 3.

2-applicaties te bouwen vanuit Visual Studio 2008. We gaan onze eerste 2-applicatie bouwen met Visual Studio 2008. Kies hiervoor 'Silverlight Application' in de 'New Project'-window. Visual Studio geeft ons nu de mogelijkheid meteen een nieuwe website te genereren, waarin we onze applicatie kunnen testen (zie afbeelding 1). De solution bevat uiteindelijk twee projecten: het Silverlight en de test-website.

Silverlight 2 steunt op de scheiding van opmaak en code: de opmaak wordt gebouwd in XAML, code-behind kan worden geschreven in VB of C#. Code-behind die gekoppeld is aan een XAML-file zal worden gedownload en uitgevoerd op de client. In codevoorbeeld 1 is te zien hoe deze koppeling tot stand komt: het attribuut 'x:Class' zorgt dat een instantie van 'HelloWorld.Page' zal worden aangemaakt bij het laden van de pagina. Deze kun je terugvinden in code-behind, te zien in codevoorbeeld 2.

Om deze code uit te voeren, zal Visual Studio de XAML en code-behind compileren naar een .NET-assembly. Deze wordt dan samen, al dan niet met andere resources, verpakt in een *.XAP-bestand (uitgesproken als ZAP). Dit is een gecompri-meerd bestand dat het ZIP-algoritme gebruikt om zo de grootte van de download te beperken. Silverlight 2 kan overigens helemaal zonder Javascript werken. Om onze eerste Silverlight-applicatie in een browserpagina te laten werken, kunnen we ofwel gebruik maken van een <object>-tag (getoond in codevoorbeeld 3), ofwel van de <asp:Silverlight>-control.

Nieuwe features in 2

Met de release van Silverlight 2 Beta 1 zijn ook heel wat nieuwe features toegevoegd. Hier zetten we de belangrijkste op een rijtje. Tot nog toe bevatte Silverlight geen controls zoals een TextBox en Button. Daarin is nu verandering gekomen: de meest gebruikelijke controls zitten ingebakken in de run-time. De overige controls, zoals ListBox, RadioButtons, een DateTimePicker en een Data-Grid, zitten in een reeks afzonderlijke *.dll-bestanden. Wanneer we deze willen gebruiken in onze applicatie, zal de assembly worden opgenomen in de XAP-file.

```
<Grid x:Name="LayoutRoot" Background="White" ShowGridLines="True">
  <Grid.RowDefinitions>
    <RowDefinition Height="30"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width=".8*"></ColumnDefinition>
    <ColumnDefinition Width=".2*"></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <TextBlock Text="Silverlight RSS Reader" Grid.Column="0" Grid.
    .Row="0"></TextBlock>
  <HyperlinkButton Click="HyperlinkButton_Click" Grid.Row="0" Grid.
    Column="1">
    <Image Source="refresh.png"></Image>
  </HyperlinkButton>
  <my:DataGrid x:Name="grdRssItems" Grid.Row="1" Grid.Column="0" Grid.
    ColumnSpan="2" AutoGenerateColumns="true" ></my:DataGrid>
</Grid>
```

Codevoorbeeld 4.

```
public Page() {
  InitializeComponent();
  string strRss = "http://www.snowball.be/SyndicationService.asmx/GetRss";
  HttpRequest request = (HttpRequest)HttpRequest.Create(new
  Uri(strRss));
  request.BeginGetResponse(new AsyncCallback(handleFeed), request);
}

void handleFeed(IAsyncResult asyncResult) {
  List<RssItem> items = new List<RssItem>();
  HttpRequest request = (HttpRequest)asyncResult.AsyncState;
  ...
  grdRssItems.ItemsSource = items;
}
```

Codevoorbeeld 5.

De aanwezigheid van een DataGrid is opvallend, aangezien er momenteel nog geen DataGrid beschikbaar is voor WPF. Naast controls zijn nu ook lay-out containers toegevoegd, zoals de Grid (niet te verwarren met de Datagrid) en de StackPanel. In een webomgeving zijn deze zeer belangrijk: relatieve positionering zorgt ervoor dat de applicatie zich kan aanpassen aan verschillende venstergroottes. In Silverlight 1.0 bestond enkel de Canvas, waarmee absolute positionering kon worden gedaan.

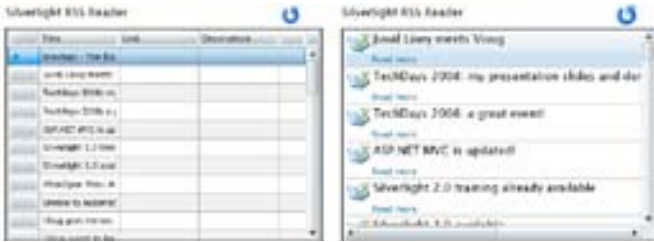
Op het vlak van security is er ook heel wat veranderd. Nog steeds draait elke Silverlight-applicatie in een 'sandbox', zodat deze geen toegang krijgt tot lokale resources en bestanden. Nieuw is wel de OpenFileDialog, waarmee een stream naar een bestand op het lokaal bestandssysteem kan worden geopend. Dit kan nuttig zijn in bijvoorbeeld een applicatie om beelden te bewerken of bestanden te uploaden. De Isolated Storage, een locatie waarin een Silverlight-applicatie zijn gegevens en data kan bewaren, is nog steeds beperkt in grootte. Wel kan nu met een prompt naar de gebruiker gevraagd worden om deze te vergroten. Ook kan Silverlight nu data ophalen van services die niet in hetzelfde domein staan. Hiervoor kunnen de WebClient- of HttpRequest-classes gebruikt worden. Deze data kunnen dan weer met DataBinding-features gebonden worden aan controls, zoals de ListBox en de DataGrid. Bovendien worden DataTemplates ondersteund, waarmee data naar wens gevisualiseerd kunnen worden. Silverlight is ontworpen met flexibiliteit van de user-interface in het achterhoofd. Het ondersteunt zowel skinning als styling. Styling is het aanpassen van de lettertypes, kleuren, enzovoort, te vergelijken met CSS. Skinning is via XAML het compleet wijzigen van het uiterlijk van een control door zelf de 'visual tree' van de control op te bouwen.

Een RSS-reader met Silverlight

We gaan een aantal van de nieuwe features gebruiken in een voorbeeld. We bouwen een RSS-reader die op een website kan worden geplaatst en de RSS-feed van een (andere) site zal uitlezen en tonen. Voor de algemene structuur van de control gebruiken we één van de nieuwe lay-outcontainers, namelijk de Grid. Een Grid is te vergelijken met een Table in html: we kunnen er rijen en kolommen in definiëren en ook RowSpan

```
<ListBox.ItemTemplate>
  <DataTemplate> <StackPanel>
    <StackPanel Orientation="Horizontal">
      <Image Source="item.png" Margin="3"></Image>
      <TextBlock Text="{Binding Title}"></TextBlock>
    </StackPanel>
    <StackPanel>
      <HyperlinkButton Content="Read more" NavigateUri="{Binding
  Link}" />
  </StackPanel>
</DataTemplate>
```

Codevoorbeeld 6.



Afbeelding 2. Links met Grid, rechts met ListBox

en ColumnSpan zijn bekend terrein voor de Grid. In codevoorbeeld 4 maken we een Grid aan, met twee rijen en twee kolommen. We kunnen absolute breedtes en hoogtes opgeven, maar ook relatieve waarden zijn toegelaten. Deze worden aangeduid met een *, bijvoorbeeld 0.8*.

Om de controls binnen de verschillende cellen te plaatsen, gebruiken we een XAML-feature: 'attached properties'. Een voorbeeld hiervan vind je in Grid.Column in het TextBlock: het TextBlock zal zich positioneren binnen de Grid in de opgegeven Column. In eerste instantie willen we de titels van een RSS-feed tonen in een DataGrid. Het DataGrid-control bevindt zich niet in de set van controls die standaard aanwezig zijn in de run-time. Om deze toch gekend te maken, wordt een reference gelegd naar 'System.Windows.Controls.Data'. Deze assembly zal mee verpakt worden in het *.XAP-bestand na compilatie. Communiceren met services (web en WCF) en data ophalen, zelfs al komen deze van een ander domein, is, zoals eerder aangehaald, in de 2-versie sterk vereenvoudigd en uitgebreid. Voor deze applicatie lezen we de RSS-feed van mijn blog uit en werken we met deze items verder. We gebruiken hiervoor de HttpRequest. Nadat alle items zijn ingelezen, wordt er databinding uitgevoerd op de DataGrid door de property ItemsSource. Op codevoorbeeld 5 wordt deze code getoond. Merk op dat alle communicatie asynchroon verloopt.

Het resultaat is de grid links in afbeelding 2. Nu we weten dat het inlezen van de data werkt, kunnen we de lay-out van de applicatie verbeteren. We vervangen de DataGrid door een ListBox. In deze laatste passen we met een custom template de manier aan waarop de items worden weergegeven. In codevoorbeeld 6 zien we hoe we de ItemTemplate overschrijven. Ook hier wordt databinding gebruikt, onder meer bij de TextBlock. Het resultaat is rechts te zien op afbeelding 2. Elk ListItem bevat nu een afbeelding, een titel en een link.

DeepZoom

We ronden af met een kleine introductie over DeepZoom. Met deze technologie wordt een grote afbeelding opgesplitst in vele 'tiles' van verschillende groottes. Naarmate de gebruiker inzoomt, wordt een andere tile-set ingeladen. Zo krijgt de gebruiker sneller de gehele afbeelding (met lagere resolutie) te zien, en bij inzoomen worden alleen de tiles ingeladen waarop de gebruiker inzoomt. Dit zorgt voor een betere user-experience en lagere kosten voor de aanbieder, aangezien alleen de delen waarvan de gebruiker meer details wil, moeten worden gedownload. DeepZoom is in Silverlight aanwezig in de vorm



Afbeelding 3. Hard Rock Memorabilia met DeepZoom

van de MultiScaleImage. In afbeelding 3 zien we DeepZoom in actie op de site van Hard Rock Cafe Memorabilia. Eerst wordt de lage resolutie-afbeelding geladen, bij inzoomen worden andere tiles ingeladen.

Besluit

Met Silverlight 2 brengt Microsoft een aantrekkelijk platform voor webontwikkelaars op de markt die op zoek zijn naar een manier om hun webapplicaties zeer interactief te maken. De belangrijkste vernieuwingen zijn natuurlijk de mogelijkheid te ontwikkelen in .NET en de toevoeging van de volledige control-set, maar ook de vele andere nieuwe features maken van Silverlight het uitgelezen platform om businessapplicaties op het web te bouwen.

Gill Cleeren is Microsoft MVP voor ASP.NET en Microsoft Regional Director. Hij werkt als softwarearchitect bij Ordina België en doet voornamelijk UI-technologieën als ASP.NET, AJAX, WPF en Silverlight. Hij heeft een passie voor nieuwe technologie en elke alpha en bèta wordt dan ook stevast geïnstalleerd. Meer lezen kan op www.snowball.be en blog.n-technologies.be.

Referenties:

Silverlight home page: www.silverlight.net

Moonlight: <http://www.mono-project.com/Moonlight>

MSDN Silverlight Dev Center: <http://msdn.microsoft.com/silverlight>

Hard Rock Memorabilia: <http://memorabilia.hardrock.com/>

(advertentie MS Press)



Programming Microsoft Robotics Studio

ISBN: 9780735624320

Auteurs: Sara Morgan

Pagina's: 288



Introducing Microsoft Silverlight 2, Second Edition

ISBN: 9780735625280

Auteurs: Laurence Moroney

Pagina's: 352