

Silverlight Tutorial

BUILDING A NETWORK-AWARE DESKTOP APPLICATION

Mike Taulty

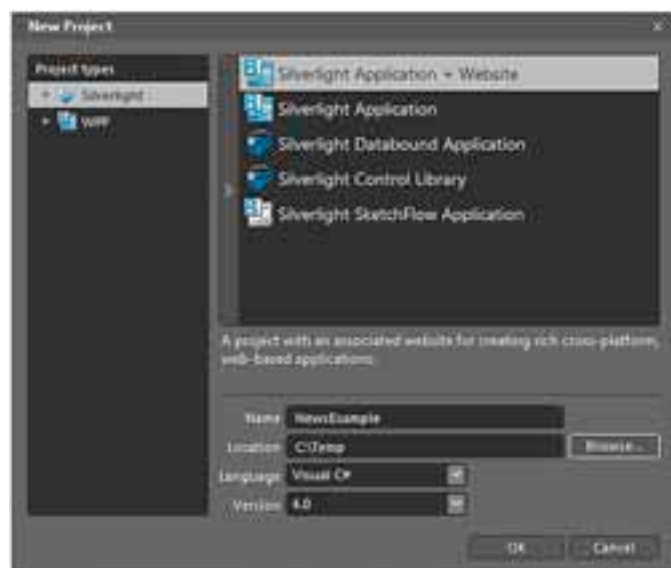
Een van de in het oog springende nieuwe features in Silverlight 3 is de mogelijkheid om applicaties een uitgedeelde lokale installatie te laten uitvoeren zodat de gebruiker in staat is deze applicatie als het ware vanaf de desktop te starten buiten de bekende browseromgeving.

Een belangrijk voordeel van deze werkwijze is dat de applicatie wel degelijk wordt gerund in exact dezelfde security sandbox, identiek aan het draaien binnen de browser, zodat richting de gebruiker het securitymodel onveranderd is. Daarnaast maakt deze techniek het de applicatie mogelijk om te detecteren of de gebruiker online of offline werkt, zodat de juiste maatregelen kunnen worden genomen rondom het gebruik van network services indien de gebruiker geen connectivity heeft.

Applicaties die een newsfeed weergeven hebben Hello World vervangen als allereerste demo, dus in deze tutorial zullen we een eenvoudige applicatie ontwikkelen die een newsfeed weergeeft en zowel binnen als buiten de browser draait. Daarnaast is deze applicatie ook op de hoogte van het feit of de netwerkkabel wel of niet is ingeplugged.

Wat heb je nodig voor deze tutorial

Als je van plan bent om de tutorial zelf te volgen, heb je de Silverlight SDK Version 3 en een trial versie van Expression Blend 3 nodig. Natuurlijk kun je ook de Release Candidate van Expression Blend 4 gebruiken. Details waar je de software kunt downloaden staan vermeld in de resources sectie aan het einde van het artikel.

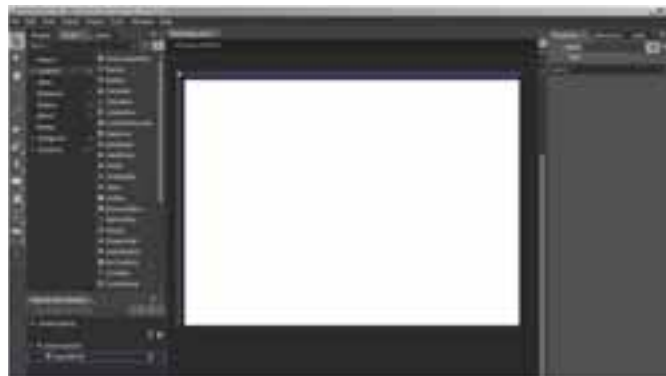


FIGUUR 1

Stap 1 – Een nieuw Silverlight Project

In Expression Blend, gebruik de File|New Project... menu optie om de New Project dialog te activeren.

Selecteer de 'Silverlight Application + Website' project template, kies een geschikte naam en folder voor je project tezamen met de implementatietaal van je voorkeur (Visual C# of Visual Basic) en klik OK om het nieuwe project aan te maken. We zullen in het resterende deel van de tutorial voor het coderen gebruikmaken van Visual C#.



FIGUUR 2

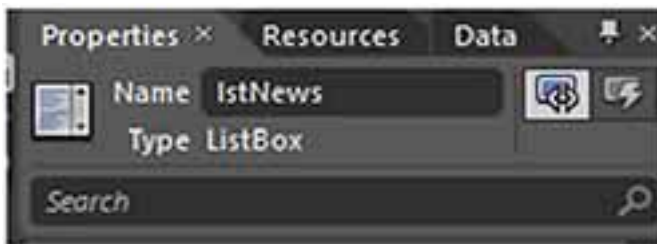
Dit lege scherm is een Grid. Klik op de shaded border links van de Grid om een nieuwe regel aan de Grid toe te voegen op ongeveer 20% van de onderkant van de grid, zoals figuur 3 laat zien.



FIGUUR 3

Nu we het Grid in 2 regels hebben verdeeld, gebruiken we het Assets panel om een ListBox control te selecteren om vervolgens een ListBox op de bovenste regel van de Grid te plaatsen. Margins zijn een zeer nuttige hulp om de ListBox goed te plaatsen binnen

de cel, een rode kleur is een indicatie als je de grenzen van de cel in de Grid benadert. Gebruik het Properties panel om de naam van de ListBox te wijzigen in lstNews zoals hieronder te zien is.



FIGUUR 4

We gebruiken het Assets panel nogmaals om een dit maal een Button te selecteren en deze button te tekenen op de onderste regel van de Grid. Gebruik het properties panel om de Content property van de Button te wijzigen, zodat deze "Get News..." aangeeft en wijzig de naam van de Button tevens in btnNews. Het resultaat is dat we nu een minimale user face hebben samengesteld, figuur 5 laat het voorlopige resultaat zien.



FIGUUR 5

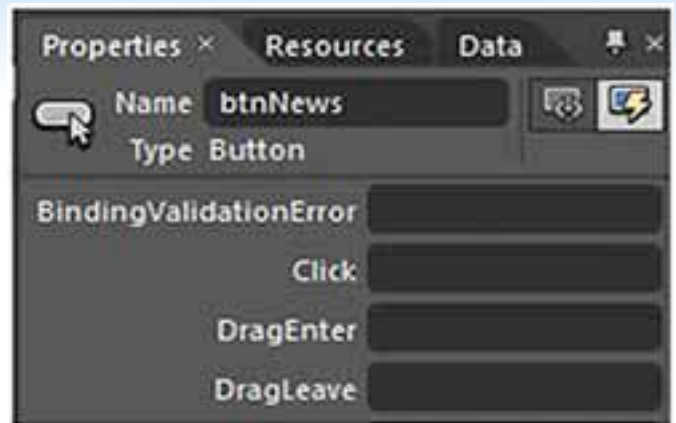
Stap 3 – Code toevoegen om de News feed te laden

We zullen enkele regels code toevoegen die wordt uitgevoerd wanneer we op de button klikken zodat we een web-based RSS feed kunnen lezen en onze ListBox kunnen vullen. Je kan gebruik maken van je eigen feed, voor dit artikel hebben we gekozen voor de feed van de Guardian WebSite die je kunt vinden op <http://feeds.guardian.co.uk/theguardian/rss>

Overigens, we hebben ervoor gekozen om de code in dit voorbeeld korte te houden, er is geen rekening gehouden met enige coding best practice.

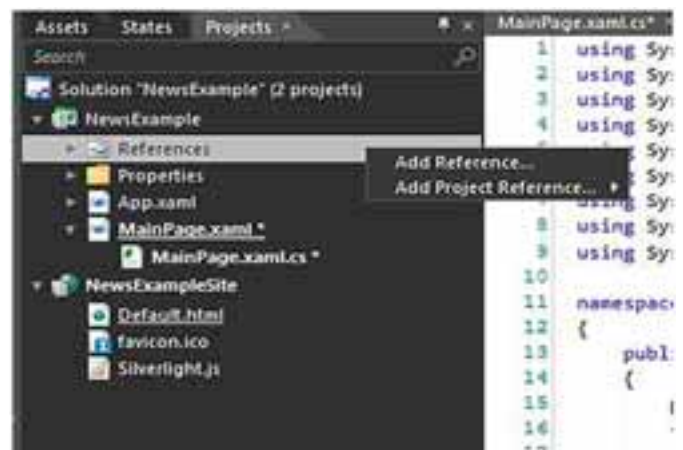
Verzeker je ervan dat de Button in de user interface is geselecteerd. Gebruik vervolgens het Properties panel om de event handlers van de button te kunnen wijzigen.

In de code voegen we using statements toe, dit maakt het gemakkelijker om de Framework classes te gebruiken.



FIGUUR 6

Dubbel-klik op de Click handler om de code editor van Expression Blend te activeren. Dit is bekend terrein voor een ontwikkelaar bekend met Visual Studio. We zijn nu zover dat we enkele regels code kunnen toevoegen die de RSS feed van de Guardian website leest. Om het ons gemakkelijk te maken hebben enkele classes nodig uit een andere .NET assembly, namelijk System.ServiceModel.Syndication.dll. In het Projects panel kunnen we deze assembly aan ons project toevoegen, klik met de rechter muis knop op de References folder, selecteer Add Reference om een referentie naar de bewuste assembly toe te voegen.



FIGUUR 7

Gebruik het dialog om te navigeren naar de equivalent van C:\Program Files\Microsoft SDKs\Silverlight\v3.0\Libraries\Client en selecteer de System.ServiceModel.Syndication.dll. Bovenaan in de code voegen we enkele using statements toe, dit maakt het gemakkelijker om de Framework classes in onze code te gaan gebruiken.

```
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.NetworkInformation;
using System.ServiceModel.Syndication;
using System.Xml;
```

We zijn zover om de code van de event handler te schrijven die we willen uitvoeren bij het klikken van de button. We zullen een routine vervaardigen die begint met een asynchrone download van de RSS feed van de website naar aanleiding van de Button klik. Het raamwerk van de routine is door Expression Blend al aan onze module toegevoegd, we behoeven ons enkel om de details te be-

kommeren. Voeg de ontbrekende regels code toe aan de routine zodat het overeenkomt met onderstaand voorbeeld.

```
private void Button_Click(object sender, System.Windows.RoutedEventArgs e)
{
    // Clear any previous values.
    lstNews.ItemsSource = null;

    // Create an object to do the download of the feed for us.
    WebClient webClient = new WebClient();

    // Give it a callback to call when the operation is completed.
    webClient.DownloadStringCompleted += OnDownloadCompleted;

    // Ask it to do the download for us.
    webClient.DownloadStringAsync(new Uri(
        "http://feeds.guardian.co.uk/theguardian/rss"));
}
```

De code gebruikt de WebClient class om een asynchronous download van de RSS feed uit te voeren. Zodra de download gereed is, zullen we een andere routine, OnDownloadCompleted genoemd, triggeren om het werk af te ronden. De onderstaande code laat de details van de routine zien.

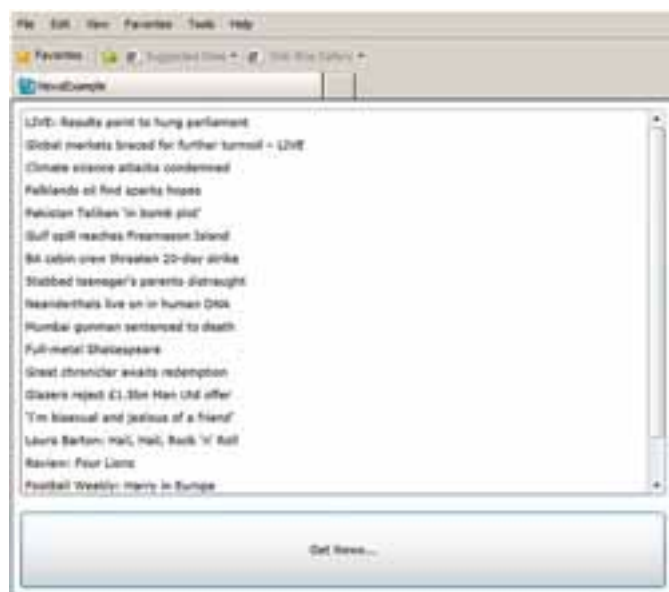
```
void OnDownloadCompleted(object sender, DownloadStringCompletedEventArgs args)
{
    // Did it seem to work?
    if (args.Error == null)
    {
        // Parse the string result as an RSS Feed.
        SyndicationFeed feed = SyndicationFeed.Load(XmlReader.Create(new
        StringReader(args.Result)));

        // Pull out the titles for our ListBox.
        lstNews.ItemsSource =
        from item in feed.Items select item.Title.Text;
    }
}
```

We willen de titels van de newsitems in de ListBox weergeven, met enkele regels code kunnen we dit realiseren.

Stap 4 – Testen van de Code

Net als in Visual Studio, vervult ook in Expression Blend de F5 key de rol om onze applicatie te starten – de user interface ziet er nu ongeveer uit als in onderstaande figuur.



FIGUUR 8

Gebruik de rechter muis knop boven de user interface, we zien dat er slechts een enkele optie wordt weergegeven, namelijk die voor 'Silverlight' informatie.

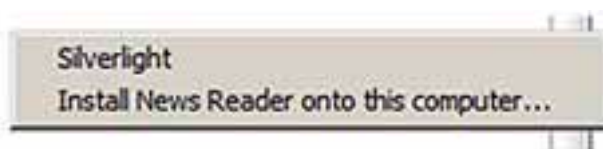
Stap 5 – De application local installeren

Nu dat de applicatie in de browser werkt, kunnen we onze focus verplaatsen en de applicatie ook beschikbaar maken op de desktop. Dit vereist een eenvoudige wijziging in de metadata door wat XML informatie toe te voegen aan een file die reeds in je project aanwezig is.

Gebruik de Projects panel om het AppManifest.xaml bestand te openen in de Properties folder. Wijzig het AppManifest.xaml bestand en voeg de Deployment.OutOfBrowserSettings sectie toe.

```
<Deployment xmlns="http://schemas.microsoft.com/client/2007/
deployment"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Deployment.Parts>
  </Deployment.Parts>
  <!-- Added this section -->
  <Deployment.OutOfBrowserSettings>
    <OutOfBrowserSettings
      ShortName="News Reader" >
      <OutOfBrowserSettings.Blurb>News Reader Example</
      OutOfBrowserSettings.Blurb>
      <OutOfBrowserSettings.WindowSettings>
        <WindowSettings Title="News Reader Example" />
      </OutOfBrowserSettings.WindowSettings>
    </OutOfBrowserSettings>
  </Deployment.OutOfBrowserSettings>
</Deployment>
```

In dit voorbeeld gebruiken we slechts een minimale configuratie, meer mogelijkheden voor custom icons en de presentatie van het window kunnen middels deze configuratie sectie worden ingesteld. Nu we deze wijziging hebben gemaakt kunnen de applicaties opnieuw runnen (F5) in Expression Blend. Zodra de applicatie zichtbaar is in de browser is een nieuwe menu optie beschikbaar onder de rechter muis menu.



FIGUUR 9

Indien we deze nieuwe optie kiezen, kunnen de applicatie lokaal installeren, de dialoog geeft ons nog enkele opties.



FIGUUR 10

Verifieer zodra de applicatie is geïnstalleerd dat deze vanaf de Desktop is uit te voeren en dat de applicatie nog correct functioneert door de RSS data op te halen door op te button te klikken. Het is ook mogelijk de applicatie wederom te verwijderen door



CONCENTRATIE TIJDENS EEN SILVERLIGHT-TRAINING.

via hetzelfde rechtermuis menu de optie “Remove this applicati-on..” te selecteren.



FIGUUR 11

Stap 6 – Toevoegen van Network Status Detectie

Het detecteren van de network status vanaf Silverlight 3 blijkt een vrij eenvoudige taak te zijn, we kunnen een property uitlezen en tevens kunnen we ook een event handler toevoegen die reageert op wijzigingen in de network status. We zullen onze code updaten om de “Get News” button van onze applicatie te activeren op basis van de beschikbaarheid van netwerk connectiviteit.

Ook hier zullen we een shortcut toepassen in de code, normaliter is het gebruik van data-binding of behaviours in Silverlight de aanbevolen manier.

Verifieer de code in het MainPage.xaml.cs bestand. Door een eenvoudige routine toe te voegen die de btnNews Button zal enablen / disablen op basis van de network status.

```
void EnableDisableButton()
{
    btnNews.IsEnabled = NetworkInterface.GetIsNetworkAvailable();
}
```

Ook nu zullen de werking valideren, echter we dienen eerst ervoor te zorgen dat de routine zal worden uitgevoerd bij het opstarten van de applicatie. We zullen de constructor van de MainPage class aanpassen zoals hieronder aangeven.

```
public MainPage()
{
    // Required to initialize variables
    InitializeComponent();

    EnableDisableButton();

    NetworkChange.NetworkAddressChanged += (s,e) =>
    {
        EnableDisableButton();
    };
}
```

We zullen wederom de applicatie in Expression Blend starten middels F5. Zodra de applicatie draait (ofwel binnen de browser of vanaf de desktop), controleer dan of de applicatie correct functioneert en de status van de button wijzigt indien de wireless network connectie wordt gestopt of de network kabel te un-pluggen.

Links

- Silverlight SDK version 3
<http://www.microsoft.com/downloads/details.aspx?FamilyID=2050e580-f1d5-4040-bb09-e6185591b6b5&displaylang=en>
- Expression Blend 3 Trial / Expression Blend 4 RC
<http://expression.microsoft.com/en-us/cc136523.aspx>

.....
Mike Taulty, werkt voor de Developer and Platform Group bij Microsoft UK, hij is actief als evangelist van het Microsoft Platform en Developer Tools. Mike is te vinden op <http://mtaulty.com>.



Microsoft + Logica
= brilliant together



Guru4Pro sessie .NET 4!

Met de introductie van .NET 4 en Visual Studio 2010 ontstaan er tal van nieuwe mogelijkheden! Kijk jij ook zo uit naar de nieuwe features? Kom dan naar Logica's Guru4Pro sessie.

Woensdag 16 juni hebben we Microsoft Guru Arie Leeuwesteijn uitgenodigd om je alles te vertellen rondom de release. Hij neemt je mee naar de nieuwe .NET wereld. Schrijf je nu in via www.logica.nl/guru4pro, zodat je zeker bent van een uitnodiging.

GURU
4PRO