

Onder ontwikkelaars is het redelijk bekend; Microsoft heeft een concurrent gemaakt voor Flash. Iets nauwkeuriger: Microsoft is de concurrentie met Adobe aangegaan door een alternatief voor de Flashplayer te maken. Dit product heet Silverlight.

Overal

Met Silverlight kunnen ontwikkelaars webapplicaties voorzien van vectortekeningen en animaties. Net als Flash is voor de eindgebruiker Silverlight een control plugin voor de internetbrowser. Ontwikkelaars gebruiken Visual Studio en de nieuwe programma's Expression Blend en Expression Design om de bestanden te maken die door de plugin gelezen worden. Deze bestanden bevatten informatie over de af te beelden tekeningen en af te spelen animaties. Interactie met de eindgebruiker is natuurlijk mogelijk net als met Flash. Als gevolg van de acties van de eindgebruiker kan de plugin andere tekeningen en animaties tonen maar ook contact zoeken met de webserver en daar data naar toe sturen of ophalen. Dit vereist enige logica die in de webbrowser moet draaien.

In versie 1.0 van Silverlight schrijft de ontwikkelaar deze programmatuur in JavaScript. Deze code wordt opgenomen in, of gelinkt aan, de pagina waar het control in staat. Door deze structuur is het mogelijk gebruikt te maken van Ajax om zo interactie met de server te hebben zonder de pagina te verversen. Het schrijven van JavaScript vind ik geen leuk werk. Als je gewend bent aan een rijk platform zoals het .NET Framework is het een flinke stap terug. Gelukkig is Microsoft het met mij eens en is er een versie 1.1 onderweg. Een van de belangrijkste verschillen tussen Silverlight 1.0 en 1.1 is dat bij 1.1 er in de plugin een mini versie van de .NET Runtime is opgenomen. Deze compacte variant van .NET kan .NET assemblies laden, compileren van IL naar machinecode en deze uitvoeren. Naast de runtime, inclusief Garbage Collector, is er ook een selectie van classes uit het .NET Framework aanwezig.

Het gevolg van deze structuur is dat de Silverlight-plugin niet alleen tekeningen en animaties meekrijgt om af te beelden maar ook overweg kan met een .NET DLL. Ik kan in Visual Studio, gebruikmakend van een prettige programmeertaal en een rijk platform logica schrijven die gedraaid zal worden op de PC van de eindgebruiker, zonder dat deze PC over het .NET Framework of zelfs Windows hoeft te beschikken. De plugin is voldoende. Natuurlijk draait de code in een browser sandbox. Wat het coderen betreft voelt een beetje als een Java Applet. De codenaam van Silverlight was WPF/E. WPF is Windows Presentation Foundation, een rijk platform voor het bouwen van Windows Forms applicaties met behulp van vectortekeningen en animaties. De E stond voor *Everywhere*, overal. Ik heb net NET 3.5 geïnstalleerd op mijn PC; een download van 50 MB. Met het uitkomen van .NET in 2001 was het nog de vraag hoe snel een redelijk aantal computers het .NET Framework aan boord zou hebben. Nu heb ik al drie versies naast elkaar staan, de Silverlight-plugin niet eens meegerekend. Als eindgebruiker hoef ik me hier helemaal niet druk over te maken. Als ontwikkelaar wordt het .NET Framework meer en meer het programmeerplatform dat ik overal gebruik.

Moest ik voor .NET bij elke programmeertaal, die ik op het Windows platform gebruikte, weer een nieuwe API leren; Visual Basic, Turbo Pascal, PowerBuilder en de Windows API voor C en C++, sinds de komst van .NET is dat veel minder het geval. Een van de beloften uit juli 2000 toen .NET werd aangekondigd wordt hiermee, langzaam maar zeker, ingelost. Nu nog even .NET 3.5 leren.

Erno de Weerd

Info Support. Over deze column kan verder gediscussieerd worden op <http://blogs.infosupport.com/ernow>