

# Ontwikkelaars en designers werken gezamenlijk aan project

EXPRESSION BLEND LUIDT NIEUWE FASE IN ONTWIKKELING  
WINDOWS-APPLICATIES

Met de komst van Windows Vista en Windows Presentation Framework wil natuurlijk iedereen strakke en effectieve gebruikersinterfaces bouwen. Microsoft Expression Blend is een ontwerpomgeving voor het opzetten en uitwerken van next-generation WPF-user-interfaces. In dit artikel maken we kennis met de mogelijkheden.

Met de Expression-lijn zet Microsoft een serieuze stap in de markt voor creatieve professionals. De Expression-familie bestaat uit Expression Web (vergelijkbaar met Adobe Dreamweaver of Microsoft FrontPage), Expression Design (vergelijkbaar met Adobe Fireworks of Adobe Photoshop/Illustrator) en Expression Blend. Van deze drie is Expression Blend de interessantste. Met dit product wordt een nieuwe fase in de ontwikkeling van het Windows-platform ingeluid. Het biedt naast de traditionele zakelijke toepassingen namelijk ook vele mogelijkheden voor creatieve en mediageoriënteerde toepassingen. We hebben het hier uiteraard over Windows Vista en de nieuwe technologie die daaraan ten grondslag ligt, in het bijzonder WPF, het Windows Presentation Framework.

In het kort valt Expression Blend het best te omschrijven als een ontwerpomgeving voor het opzetten en uitwerken van next-generation WPF-user-interfaces. De kern van WPF ligt in de taal XAML, de eXtensible Application Markup Language. Waar webpagina's beschreven met CSS en html, ondersteund door serverside-logica en clientside-logica in Javascript, daar wordt een WPF-applicatie beschreven door XAML en ondersteund door logica in code-behind files. Maar daarmee houdt iedere overeenkomst verder ook op. XAML is een geheel nieuw ontwikkelde taal op basis van XML, waarmee je de interface van een applicatie declaratief kunt beschrijven. Hiervoor staat een uitgebreide bibliotheek WPF-componenten ter beschikking met niet alleen de gebruikelijke elementen zoals buttons en labels, maar ook flexibele lay-outcontainers die een *constraints-based lay-out* ondersteunen. Dit alles wordt ondersteund door een krachtige render- en compositing-engine, die op eenvoudige wijze effecten als transparantie, schaling, rotatie, masking mogelijk maakt.

Maar het zijn niet alleen visuele effecten die WPF en Expression Blend bijzonder maken. De technische architectuur van WPF en de opzet van Blend maken een omslag in de manier van werken mogelijk. Tot voor kort was het ontwikkelen van een Windows-applicatie eigenlijk alleen voorbehouden aan ontwikkelaars met ervaring in Visual Studio. Daarmee is het experimenteren met gedurfde of aantrekkelijke user-interfaces, die verder gaan dan de standaard Windows-applicatie, een bewerkelijke aangelegenheid. Het betekent een langdurig touwtrekken tussen ontwerpers en ontwikkelaars. De subtiele details

die een fraaie interface van een slecht vormgegeven interface onderscheiden, zijn daarvan vaak het eerste slachtoffer. Ook voor ontwikkelaars is dit geen prettig proces. Het bouwen van interfaces gaat in de Visual Studio-designer wel een stuk prettiger omdat je niet alles handmatig met code hoeft te typen. Maar de Visual Studio designer moet uiteindelijk wel gewoon procedurele code genereren en deze voorzichtig tussen de overige andere code in plaatsen en dat gaat ook wel eens mis. Bovendien is het heen en weer slepen van buttons en tekstboxen doorgaans niet het leukste werk.

## Developers en designers

XAML en Expression Blend maken een veel natuurlijker manier van werken mogelijk. Zo kunnen ontwerpers met Blend Windows-applicaties ontwerpen die qua interface vrijwel grenzeloos gevarieerd zijn. Omdat de code in de code-behind files gescheiden is van de interfacedefinitie in de XAML-files kunnen de ontwikkelaars los van de ontwerpers aan de functionaliteit van de applicatie werken. De ontwikkelaars hoeven dus niet bang te zijn dat een ontwerper per ongeluk fouten introduceert in de code. Omgekeerd hebben de ontwerpers een veel grotere mate van vrijheid en controle bij het ontwerpen van de interface. Het ontwikkelproces wordt beter schaalbaar en minder vatbaar voor fouten in communicatie of interpretatie.

Is Blend dan vooral geschikt voor ontwerpers? Ja en nee. Blend is deels duidelijk gericht op de creatieve markt. Wanneer we Blend voor de eerste keer opstarten, presenteert het zich heel nadrukkelijk als een grafisch en animatiepakket. Het bevat elementen die voor de gebruikers van Flash of Photoshop meteen herkenbaar zijn, zoals de toolbar met de gebruikelijke (grafische) tools links in het scherm, een royaal ontwerpvenster in het midden waar via een tab geschakeld kan worden tussen de Design-view en de XAML-code, een overzicht van 'layers' onder in het scherm en de property-panels rechts. Waar eerdere versies van Blend (toen nog Expression Interactive Designer geheten) zich als een conventioneel, enigszins saai, standaard Windows-programma presenteerden, heeft Blend nu een eigen, industriële antracietgrijze look. Het is even wennen. Zie afbeelding 1.

Dat duurt echter niet lang. De interface is in alle opzichten een verbetering ten opzichte van de oudere versies. Iedereen die al eens met een programma als Photoshop heeft gewerkt, zal ook



Afbeelding 1. De industriële antracietgrijze look van Blend

gaw met de grafische tools van Blend overweg kunnen. De gangbare tools voor het neerzetten van gradients en Bezier-curves werken gewoon zoals verwacht - met hier en daar een leuke verrassing. Zo worden de Bezier-curves allemaal netjes omgezet in eenvoudige coördinatenreeksen in de XAML (gemakkelijk voor het knippen en plakken). Al snel blijkt dat onder het grafische huidje een krachtige tool voor ontwikkelaars schuilgaat. Zo maakt Blend gebruik van dezelfde projectfiles als Visual Studio. Een Blend-project kan dan ook probleemloos geopend worden in Visual Studio of andersom. En beide programma's kun je ook gelijktijdig openen met hetzelfde project. Wanneer wordt geschakeld tussen Visual Studio (of de Express-editie) en Blend detecteren de programma's eventuele wijzigingen vanuit het andere programma vanzelf en worden deze eenvoudig overgenomen. Blend integreert ook in andere opzichten met Visual Studio. Tekstfiles zoals code-behind files worden automatisch geopend in Visual Studio en als een nieuwe eventhandler wordt aangemaakt, wordt het geraamte hiervoor automatisch aan de code-behind toegevoegd en in Visual Studio geopend.

Nu is het één ding om snel iets moois in elkaar te zetten. Iedere ervaren ontwikkelaar weet dat uiteindelijk de meeste tijd gaat zitten in het onderhouden en aanpassen van een toepassing. Daarom is het van levensbelang om in een applicatie de inhoud, logica en presentatie zoveel mogelijk van elkaar te scheiden en een modulaire opzet te hanteren. Blend komt hieraan tegemoet via een aantal krachtige hulpmiddelen. Zo kunnen interface-elementen eenvoudig als templates of resources worden gedefinieerd en toegekend. Om de look-and-feel van de applicatie te wijzigen, hoef je alleen maar de desbetreffende resource aan te passen. Het is zo een koud kunstje om een applicatie van een nieuwe skin te voorzien.

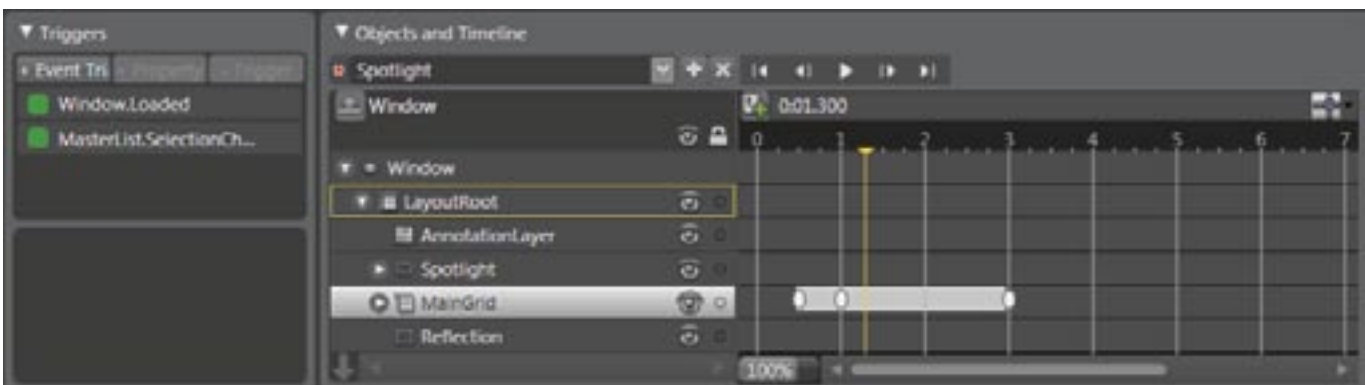
Ook is het erg eenvoudig om via Control-libraries eigen componenten op te bouwen op basis van bestaande componenten.



Afbeelding 2. Het maken van een Brush

Deze componenten kunnen dan weer van nieuwe functionaliteit worden voorzien door middel van code-behinds. Het mooie is dat deze zelfgemaakte componenten volwaardig meedraaien in Blend. Door middel van reflection haalt Blend zelf alle properties uit de component en presenteert deze aan de gebruiker in de standaard interface. Definiëren we bijvoorbeeld een Clock-component met een property BackgroundColor, dan zal deze property in de Blend-interface verschijnen en kan de gebruiker via een color-picker eenvoudig een kleur kiezen. Maar we kunnen er ook een Brush van maken. Dan heeft de eindgebruiker ineens alle Brush-mogelijkheden (zie afbeelding 2) tot zijn beschikking: van video tot een reflectie van een andere component. De mogelijkheden zijn legio. Doordat het zo gemakkelijk is om custom componenten te maken en er sprake is van een naadloze integratie van deze in de Blend-interface, wordt het bouwen van herbruikbare interfacecomponenten ook voor kleinere projecten de moeite waard.

Gaan we nog een stapje verder en maken we van de BackgroundColor-property een DependencyProperty, dan kunnen we meeliften op de Data Binding- en Animation-infrastructuur van WPF. We kunnen dan de property in Blend animeren of de waarde van de property aan een datasource koppelen. Willen we bijvoorbeeld de BackgroundColor van onze Clock-component aan de waarde van de AEX-index koppelen, dan kunnen we dit met slechts een paar extra regels code (voor het converteren van de koers naar een kleurwaarde) bewerkstelligen. Het leuke is dat we dit dan ook in Blend direct zien gebeuren. Dat



Afbeelding 3. De interface om animaties op te zetten is nog niet uitontwikkeld

wil zeggen, de Clock-component die we in de Design-view van Blend zien, is precies dezelfde Clock-component die in de uiteindelijke applicatie draait. Er is dus geen scherp onderscheid tussen 'Design'- en 'Run'-views.

Blend is nog volop in ontwikkeling. Een aantal kanttekeningen bij de eerste bètaversie is in de tweede bètaversie al verholpen, maar een aantal zaken zijn nog vatbaar voor verbetering. Zo is de interface van Blend om animaties op te zetten (zie afbeelding 3) nog niet uitontwikkeld. Er moet nog te vaak worden teruggegrepen op de XAML code view om de puntjes op de i te zetten. Daarnaast bevat Blend nauwelijks debugging-functionaliteit, zelfs geen vangnet voor niet afgehandelde exceptions. Wanneer een exception optreedt in een applicatie die vanuit Blend wordt gestart, dan sluit de applicatie af met de standaard Windows-foutmelding. Het zou handig zijn wanneer Blend deze afving en meer informatie gaf over de exception die is opgetreden. Dit is met name belangrijk omdat WPF-applicaties in de standaardconfiguratie (vanuit veiligheidsoverwegingen) niet vanaf een netwerk-share kunnen draaien. Als een beginnende gebruiker dus een project opent en bouwt vanaf een netwerk-share, dan resulteert dat in de weinig behulpzame foutmelding dat het programma onverwacht is beëindigd.

Maar uiteindelijk schuilt achter het eigzinnige uiterlijk van Blend een krachtig framework dat zowel gericht is op het snel opleveren van spectaculair 3D-audio/video geweld als op het beheerbaar houden van een project. Bovendien zit Blend zo in elkaar dat het bijna automatisch verstandige conventies als de scheiding van inhoud, lay-out, presentatie en logica aanmoedigt. Hierdoor is Blend veel meer dan een veredeld tekenprogramma of een WYSIWYG-editor, maar een onmisbaar stuk gereedschap voor de ontwikkeling van een nieuwe generatie interactieve, mediageoriënteerde toepassingen.

**Pascal Haakmat** is werkzaam als Technical Manager bij Lightmaker, hij is te bereiken via e-mail op [pascal.haakmat@lightmaker.com](mailto:pascal.haakmat@lightmaker.com)

#### Referenties

[www.microsoft.com/design](http://www.microsoft.com/design)

[www.microsoft.com/expression](http://www.microsoft.com/expression)

<http://blogs.msdn.com/expression/>

( advertentie Microsoft Press )



**Microsoft VBScript Step by Step**

ISBN: 9780735622975

Auteur: Ed Wilson

Pagina's: 512