

# Windows 7 voor developers

## PROFESSIONAL DEVELOPER CONFERENCE

Dennis van der Stelt en Alex Thissen

Op de Professional Developer Conference in Los Angeles liet Microsoft nieuwe en aankomende technologieën zien aan developers. Alex Thissen en Dennis van der Stelt waren aanwezig en hebben zich vanuit Class-A onder andere geconcentreerd op Windows 7 en de uitbreidingen die deze biedt aan developers.

Met Windows 7 zet Microsoft een platform neer om eenvoudig applicaties op te bouwen met een groter gebruiksgemak dan voorheen. Veel features uit Windows Vista zijn verbeterd en uitgebreid, maar ook toegankelijk voor developers. In dit artikel bespreken we onder meer de Ribbon-control voor developers, de mogelijkheden voor externe sensoren, de vernieuwingen in Search en het bouwen van met de hand te bedienen applicaties.

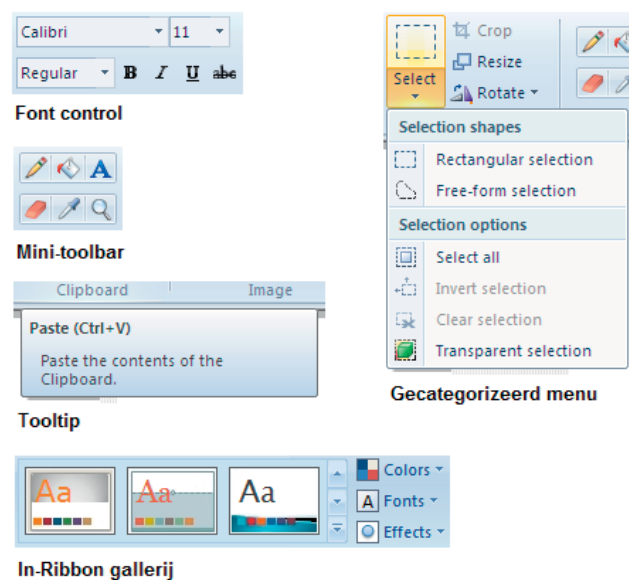
Microsoft Office 2007 introduceerde de Fluent UI, waar de Ribbon-control onderdeel van uitmaakt. Windows 7 bevat ook een 'Scenic' Ribbon-control, die nagenoeg identiek is aan die van Office. Het heeft vrijwel dezelfde features en een paar kleine aanpassingen in het uiterlijk. De aanpassingen zijn gemaakt om de Ribbon-control beter te laten passen in de Windows UI-omgeving, zoals een vierkant applicatiemenu in plaats van de ronde vorm van de Office Ribbon.

Als grote voordeel van de Windows Scenic-Ribbon geldt dat de versie in alle applicaties inzetbaar is en niet gebonden is aan Office. Iedere applicatie kan daardoor profiteren van de voordelen van de Ribbon-control, zoals een makkelijker te leren UI door de intuïtieve indelingen en rijkere presentatievormen.

De architectuur van de Scenic-Ribbon lijkt sterk op die van Windows Presentation Foundation. Er is een scheiding van verantwoordelijkheden tussen het gedrag van de Ribbon en de acties van de applicatie op basis van handelingen van de gebruiker. De Ribbon gebruikt XAML markup voor het definiëren van de layout. Via de COM API voer je de initialisatie van de Ribbon uit met behulp van de markup en vang je events af vanuit de Ribbon. In de event handlers spreek je tenslotte je applicatielogica aan. Net als in WPF is het gedrag van de controls ontkoppeld van het uiterlijk. De visuele stijl van de controls kun je daardoor automatisch baseren op de locatie binnen de Ribbon.



AFBEELDING 1. DE NIEUWE WINDOWS SCENIC RIBBON

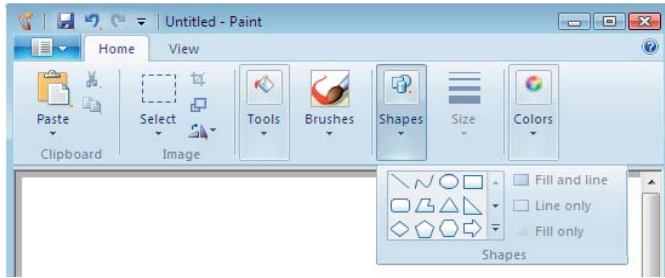


AFBEELDING 2. VOORBEELDEN VAN RIBBON-SPECIFIEKE CONTROLS

De Ribbon-control maakt gebruik van een eigen set compleet nieuwe controls. Sommige controls komen bekend voor, zoals de checkbox, spinner controls en tooltips. Daarnaast zijn er nieuwe Ribbon-specifieke controls, waaronder de 'In-Ribbon'-gallerij, de mini-toolbar, de group dialog launcher en de font control. Afbeelding 2 geeft een overzicht van een aantal controls.

De lay-out van de controls in de Ribbon-groepen past zich automatisch aan op basis van de beschikbare ruimte die de Ribbon heeft. Wanneer de controls bijvoorbeeld niet meer passen in een groep, zullen deze niet meer direct zichtbaar zijn in de Ribbon. Je vraagt ze op via een uitklapbaar menu.

De Ribbon ondersteunt verscheidene zelf te definiëren applicatiemodi. Afhankelijk van de modus zal de Ribbon tabbladen activeren of deactiveren, waardoor diverse lay-outs ontstaan. Een andere voorziening van de Ribbon is de mogelijkheid automatisch previews te krijgen van nog uit te voeren commando's. Zo krijg je bij het bewerken van tekst in een document met de font control een beeld hoe de aangepaste tekst er uit gaat zien.



AFBEELDING 3. AUTOMATISCHE LAY-OUT MANAGEMENT

## Versies van de Ribbon

De Windows Scenic Ribbon is een COM control die ter beschikking staat vanuit de Win32 API van Windows 7. De Ribbon is dus in eerste instantie bedoeld voor programmeurs van native code-applicaties. WinForms-applicaties kunnen op termijn gebruik gaan maken van een managed wrapper voor deze COM control.

Het MFC Framework heeft al sinds Visual Studio 2008 SP1 een eigen Ribbon-control die er net zo uitziet als de Office 2007-versie. Er verschijnt een update voor de MFC Ribbon in 2009 met daarin ook de Windows 'Scenic' Ribbon look-and-feel. Windows Presentation Foundation krijgt eveneens een eigen Ribbon-control met zowel het Office 2007- als het Scenic-uiterslijk. De WPF-versie heeft als voordeel dat de rendering via vector graphics loopt. Hierdoor profiteert het van alle grafische voordelen van WPF, zoals schalen en roteren. De WPF Ribbon komt in de loop van 2009 uit.

	Windows	Office 2007	Wanneer	Target code
MFC Ribbon	✓ (in 2009)	✓	VS 2008 SP1	MFC native
WPF Ribbon	✓	✓	2009	Managed
Scenic Ribbon	✓		Windows 7	Win32 native
				WinForms (via interop)

De Microsoft-teams van WPF, MFC en Windows werken nauw samen om deze drie verschillende Ribbon-versies zoveel mogelijk op elkaar af te stemmen. Ze willen hiermee bereiken dat er een consistente manier van ontwikkelen komt, ook al bestaan er drie verschillende implementaties. Zo gebruikt het WPF Ribbon al dezelfde XAML als de Windows 7-versie.

De Ribbon-control biedt veel gebruiksgemak en functionaliteit, waarvoor minimaal programmeren vereist is. De XAML markup zorgt voor een declaratieve manier van definiëren van de Ribbon-onderdelen. De Ribbon zelf regelt de rendering onder verschillende omstandigheden. Je hoeft alleen code te schrijven voor het afhandelen van de commando's en de daarop volgende applicatie-logica.

## Sensor & Location Platform

Microsoft heeft een nieuw driver-model ontwikkeld dat het in Windows 7 zal introduceren. Hiermee kunnen ontwikkelaars eenvoudiger en eenduidiger met externe sensoren werken. Als je met externe sensoren hebt gewerkt, zoals een GPS, ken je de nadelen daarvan in Windows. Bijvoorbeeld dat elke keer als je de GPS opnieuw hebt aangesloten, er een andere COM-poort in gebruik kan zijn. Of wellicht had een andere applicatie reeds je

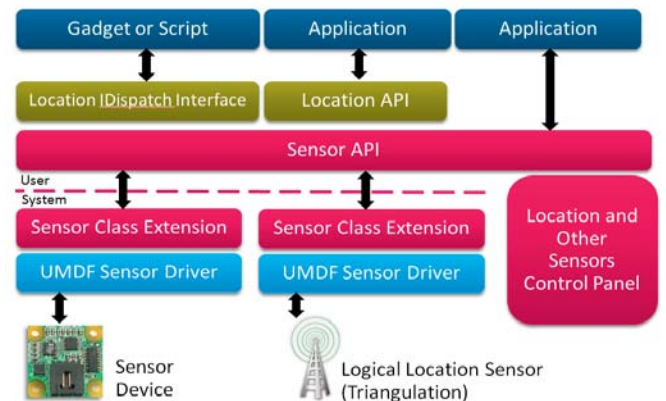
GPS toegeëigend, waardoor jij de GPS niet meer kon gebruiken. Windows 7 belooft veel van deze problemen te verhelpen.

Het nieuwe driver-model is generiek voor alle soorten sensoren en de fabrikant kan eenvoudig aan het model voldoen. De UMDF Sensor Driver (een nieuw in Vista geïntroduceerd driver-platform) wordt door de fabrikant geleverd en hij kan de Sensor Class Extension bijleveren. Deze neemt een groot deel van communicatie met de API voor zijn rekening, zoals properties, events en het formaat van de door te sturen gegevens. De developers communiceren met de sensor-API, die uit native C++/COM-code bestaat. Uiteraard bevat de SDK ook de mogelijkheid om vanuit managed .NET-code te communiceren.

De voordelen van het nieuwe driver-model zijn dat we niet meer een bepaalde COM-poort hoeven te gebruiken en dat meer applicaties tegelijk met de sensoren kunnen praten. Daarnaast heb je de mogelijkheid door bekende sensoren te zoeken op basis van categorie (wat meet de sensor) of type (hoe meet de sensor). Ook is eenvoudig uit te zoeken wat er exact gecommuniceerd dient te worden van en naar de sensoren. Uiteraard heeft men ook aan beveiliging gedacht, want de gebruiker hoeft niet altijd toe te stemmen om gegevens te delen met applicaties. Per gebruiker kan je bepalen welke sensoren standaard aan of uit moeten staan.

Bovenop de sensor-API ligt de locatie-API. Deze laatste is voor een specifiek domein, namelijk het bepalen en werken met locatiegerelateerde informatie. Zoals in afbeelding 4 te zien, ligt deze bovenop de standaard sensor-API en is deze te gebruiken om een eenvoudige vraag te stellen, zoals: "Waar bevind ik me op dit moment?". De API kan deze vraag in Windows 7 bijna net zo eenvoudig beantwoorden, hoewel dat technisch niet zo eenvoudig is. Wat gebeurt er als je tijdens de keynote op de PDC geen verbinding krijgt met satellieten? Windows 7 biedt hiervoor in het configuratiescherm de mogelijkheid een standaard locatie in te voeren. Ook nieuw in Windows 7 is dat sensoren via triangulatie kunnen bepalen waar de gebruiker zich ruwweg bevindt. Dit gaat momenteel ook al via minimaal drie WiFi-hotspots of telefoonmasten. Maar de locatie-API bepaalt zelf de meest nauwkeurige locatie aan de hand van de huidige mogelijkheden.

De locatie-API biedt standaard geografische informatie zoals breedte- en lengtegraad, plus voor de gebruiker leesbare informatie zoals land, postcode of straatnaam. Deze informatie kun je toepassen in je eigen applicatie om leesbare informatie te tonen



AFBEELDING 4. SENSOR DRIVER MODEL ARCHITECTURE

aan de gebruiker. Je sensoren en bijbehorende drivers moeten deze informatie dan uiteraard wel bieden.

Op de PDC kregen bezoekers bij de Windows 7-stand de Software Development Kit (SDK) mee, inclusief bijbehorende printplaat; bijzonder speelgoed voor de gadget-liefhebbers onder ons. Op de printplaat zitten namelijk sensoren voor licht, twee rijen met knoppen en een 3D-versnellingsmeter (accelerometer). Daarnaast een kleine buzzer en een lichtkrant, waarop tekst voorbij komt rollen die je kunt aanpassen. Op de PDC werd de SDK getoond met enkele aanpassingen aan de MSDN Reader, een WPF-applicatie waarvan de sourcecode te vinden is op de Microsoft Code Gallery. Als voorbeeld toonde men de situatie dat je met je laptop jezelf verplaatst van kantoor naar het strand van Santa Monica, waar het de gehele week rond de 30 graden was. In de zon zijn de MSDN-artikelen lastig te lezen. De aanpassing in de MSDN Reader was gebaseerd op de lichtsensoren op de printplaat. Op kantoor wordt het standaard font gebruikt, dat goed valt te lezen. Kom je in fel licht, dan past de applicatie de grootte van het font aan wat alles beter te leesbaar maakt. Uiteraard valt er meer te doen, zoals gekleurde diagrammen in zwart-wit tonen of andere contrastkleuren kiezen. Of om juist tijdens de PDC-keynote de helderheid te verlagen om de accu te sparen.

Persoonlijk vinden we de 3D-snelheidsmeter en de lichtkrant erg interessant. We beschouwen het als een persoonlijke uitdaging Space Invaders te bouwen in de lichtkrant, de tank met de snelheidsmeter te laten verschuiven en ze met de knoppen uit de lucht te schieten. Als iemand een creatief idee heeft hoe we de lichtsensoren hierbij kunnen gebruiken, horen we dat graag.

## Multi-touch support in Windows 7

De volgende generatie user experiences gaat ongetwijfeld meer gebruikmaken van multi-touch. Het gaat om de logische opvolger van de stylus voor de Tablet PC's. Microsoft introduceerde multi-touch al met Surface, de tafel met een aanraakgevoelig oppervlak. Windows 7 biedt de ondersteuning van Touch in het operatiesysteem aan. Hierdoor kunnen we alle applicaties touch-voorzieningen geven.

Touch is in applicaties op verschillende manieren in de bediening te gebruiken. In de meeste gevallen passen we touch toe voor het beter en makkelijker consumeren van en werken met informatie. Denk daarbij aan het navigeren op het web en werken met bestanden, bekijken van foto's en spelen van eenvoudige spelletjes. Multi-touch maakt gebruik van gebaren (gestures) en aanraking (touch). De gestures zijn bekende gebaren die herkend worden, zoals het maken van een neerwaartse beweging met twee vingers voor het scrollen van een venster. Touch geeft toegang tot een lager niveau van het aanraken van een touch-oppervlak en maakt het onder andere mogelijk langdurige aanraking af te handelen of nieuwe gebaren samen te stellen.

Het multi-touch team heeft een drietal gradaties van ondersteuning voor multi-touch gedefinieerd:

### 1. Goed

Iedere applicatie gaat (automatisch) gebruikmaken van standaard touch-functionaliteit binnen Windows 7. Er zijn eventueel kleine tweaks in de applicatie gemaakt om beter met touch

---

(Advertentie)

```

<Window x:Class="DotNetMagazine.MyFirstTouchWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Touch-enabled applicatie" Height="300" Width="300"
  t:TouchBehavior.IsTouchEnabled="True">
  <Grid>
  <Canvas>...</Canvas>
  </Grid>
</Window>

```

CODEVOORBEELD 1. XAML VAN TOUCH-ENABLED WPF WINDOW

om te gaan, zoals het aanpassen van de grootte van aanraakgevoelige delen.

2. *Beter*

Een applicatie biedt ondersteuning voor gestures en notificaties voor gestures. De applicatie herkent zelf gestures en kan ook selectief zijn in het uitkiezen van specifieke gestures.

3. *Best*

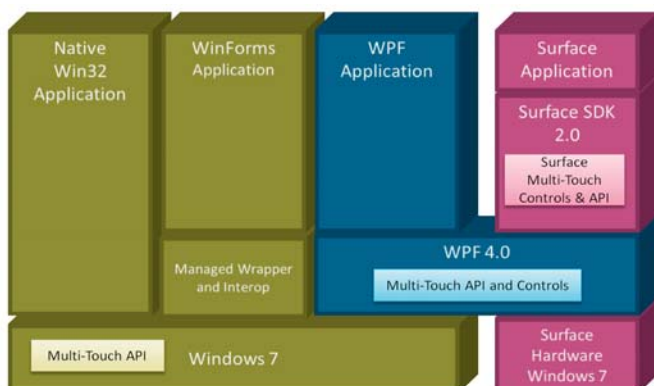
Naast gestures ook ondersteuning voor touch. De applicatie is geoptimaliseerd voor het gebruik van touch.

## Van Surface naar WPF 4.0

Windows 7 herkent al diverse gestures en delen van de Windows user interface zijn reeds geoptimaliseerd voor touch. Het zorgt voor consistente gestures bij onder andere panning en zooming in applicaties. Als een applicatie zelf geen multi-touch-ondersteuning biedt, escaleert een touch-handeling naar het operatingsysteem. Windows 7 zal het touch event vervolgens proberen toe te passen op een set van herkende controls, zoals een scroll gesture op een standaard scrollbar. We kunnen daardoor gemakkelijk een applicatie touch-enabled maken, maar het biedt nog weinig toegevoegde waarde voor de eindgebruiker.

De eerste generatie touch-applicaties op het Microsoft-platform werkt op Surface hardware met Windows Vista en gebruikt Windows Presentation Foundation 3.5 en de Surface SDK 1.0. De volgende generatie touch-applicaties zal draaien op Windows 7. Het Windows 7-operatingsysteem biedt touch-ondersteuning voor zowel Surface als WPF, Windows Forms en native Win32-applicaties.

Native Win32-applicaties maken gebruik van de multi-touch API uit Windows 7, waarin we onder andere de nieuwe windows-boodschappen WM\_GESTURE en WM\_TOUCH tegenkomen. Op dat niveau is het ook mogelijk de gesture-configuratie te veranderen, zodat je afhankelijk van de context-gestures wel of niet ondersteunt. Met gesture notificatie-boodschappen (WM\_GE-



AFBEELDING 5. OVERZICHT VAN MULTI-TOUCH-PLATFORM

STURENOTIFY) kunnen we deze configuratie zelfs net voor het versturen van de gesture-boodschappen aanpassen. Alle boodschappen leveren ruwe data die de applicatie naar eigen inzicht kan gebruiken om bijvoorbeeld nieuwe gestures te definiëren.

Met de combinatie van Windows 7 en het .NET Framework 4.0 ontstaat er een gemeenschappelijke basis voor het bouwen van WPF- en Surface-applicaties. WPF 4.0 bevat een multi-touch API en aangepaste controls die kunnen omgaan met gestures en touch. Een WPF-applicatie bouw je rechtstreeks op WPF 4.0. Surface-applicaties maken gebruik van de Surface SDK 2.0, die is gebouwd op basis van WPF 4.0. Bovendien bevat deze extra aangepaste WPF-elementen, zoals SurfaceWindow in plaats van Window.

Windows Forms maakt voornamelijk gebruik van de Win32 API via Platform Invocation services (P/Invoke). Er zijn managed wrappers, gemaakt voor touch via een real-time stylus en ink-ondersteuning.

Windows Presentation Foundation 4.0 integreert de multi-touch API door middel van aanpassingen in de basisklassen UIElement en UIElement3D. De windows-boodschappen worden afgehandeld en aangeboden als events voor gestures en touch. WPF 4.0 krijgt ook de mogelijkheid configuratie voor traagheid te zetten. Verder worden de basis-controls aangepast zodat deze in staat zijn verschillende touch-acties te vangen. Zo snapt een control bijvoorbeeld dat je met twee afzonderlijke vingers tegelijk kunt werken. De Surface SDK voegt aan WPF 4.0 nog een aantal nieuwe en Surface-specifieke WPF-controls toe. Listboxen krijgen elastische effecten, textboxes tonen na activeren een virtueel toetsenbord en meer gebruikers mogen tegelijk Surface-menu's bedienen.

## Traagheid en manipulaties

Multi-touch kent ook traagheid (inertia). Wanneer we bijvoorbeeld een foto verslepen met behulp van touch en deze snel naar een zijkant verplaatsten, glijdt de foto bij het loslaten nog een tijdje door! Dit levert een vrij natuurlijk gedrag op van de objecten

```

public partial class PhotoUserControl : UserControl
{
  private MatrixTransform originalMatrix = null;
  public PhotoUserControl()
  {
    InitializeComponent();
    this.RenderTransform = new MatrixTransform();
    this.TouchGestureStarted +=
      new EventHandler<TouchGestureStartedEventArgs>(OnGestureStarted);
    this.TouchGestureChanged +=
      new EventHandler<TouchGestureChangedEventArgs>(OnGestureChanged);
  }
  private void OnGestureStarted(object sender, TouchGestureStartedEventArgs e)
  {
    originalMatrix = ((MatrixTransform)((UIElement)sender).RenderTransform).Matrix;
    e.Handled = true;
  }
  private void OnGestureChanged(object sender, TouchGestureChangedEventArgs e)
  {
    ((UIElement)sender).RenderTransform = e.CalculateMatrixTransform(originalMatrix);
    e.Handled = true;
  }
}

```

CODEVOORBEELD 2. CODE-BEHIND VAN USER CONTROL MET TOUCH EVENT HANDLING

---

## De feed is uit te breiden met informatie over thumbnails en metadata zoals copyright, datum, enzovoort

---

waarmee een eindgebruiker bezig is. Als ontwikkelaar hoef je niet al te veel rekening te houden met deze traagheid. De onderliggende processor voor traagheid sturen nog WM\_TOUCH-boodschappen door nadat de fysieke aanraking al is afgelopen. Het afhandelen van deze boodschappen werd toch al uitgevoerd en loopt nu iets langer door. Er is dus weinig extra programmeerwerk voor nodig om traagheid te ondersteunen in je applicatie.

De laatste voorziening in multi-touch zijn manipulaties: langdurige gestures, die bovendien van de ene in de andere gesture kunnen overgaan. Manipulaties vormen een hogere abstractie van de touch API en controls en een superset van alle ondersteunde gestures. Ook voor manipulaties heeft Windows 7 een processor die achter de schermen de berekeningen uitvoert. Een applicatie die manipulaties en/of traagheid inzet, biedt de eindgebruiker een andere gebruikerservaring en valt daarmee al snel in de categorie 'best' van touch-ondersteuning.

### Touch enabled applicatie

Om een indruk te geven van het inzetten van touch in een applicatie laten we wat markup en code zien van een WPF-applicatie voor het tonen van foto's. De foto's zijn te verplaatsen, schalen en

roteren met gestures. De applicatie bestaat uit een container, in dit geval een Window, die foto's toont via user controls. We moeten de container allereerst touch-enabled maken (codevoorbeeld 1).

Zodra een container touch-enabled is, kunnen kinder-elementen gesture en touch events afvangen. Codevoorbeeld 2 toont hoe we een user control geroteerd, verplaatst en geschaald maken door het toepassen van een matrixtransformatie, die deze drie manipulaties combineert.

Uiteraard kun je touch naar eigen inzicht inzetten in je applicatie. WPF heeft daarvoor allerhande bubbling events zoals TouchDown en -Up, TouchMove, TouchEnter en -Leave, plus alle tunnelende preview events. Het werken met touch lijkt dan sterk op het afhandelen van muis-events.

### Multi-Touch

Het inzetten van Multi-Touch in je applicatie moet doordacht gebeuren. Een eindgebruiker zal op een andere manier interactie hebben met een applicatie die touch-ondersteuning biedt. Zo moet je gebieden die een gebruiker aanraakt groter maken en je dient ook rekening te houden met de grootte van het scherm. De

---

(Advertentie)

gestures moeten natuurlijk en logisch zijn om een gebruiker er efficiënt en intuïtief mee te laten werken.

Een Surface-applicatie is nog specialer. Meer personen kunnen plaatsnemen aan alle zijden van de Surface-tafel. Dit betekent dat de oriëntatie van de UI voor ieder anders is. Gebruik van bijvoorbeeld veel teksten wordt daarmee lastiger, want er zal altijd wel iemand teksten ondersteboven moeten lezen. Bovendien gaan verschillende personen tegelijkertijd interacteren met het touch-opppervlak. De applicatie moet hiermee kunnen omgaan. We verwachten een nuttig gebruik van touch in applicaties. Het biedt talloze nieuwe mogelijkheden voor de user interface. De komende jaren komen er ongetwijfeld veel innovaties op het gebied van touch.

## Search & Visualization

Met de komst van Windows Vista veranderde er al veel aan het uiterlijk van Windows Explorer. Daarbij zijn enkele wijzigingen bijzonder gebruiksvriendelijk. In Windows 7 vinden we opnieuw verbeteringen, onder meer het sneller vinden van documenten, muziek en films op je harddrive. Je kunt bijvoorbeeld metadata uit je documenten gebruiken om te sorteren op auteur. Een andere feature is het samenvoegen van muziek van verscheidene locaties op je drive in een bibliotheek. Op die manier bepaal je zelf op welke harddrive je muziek bewaart. Windows Explorer en Media

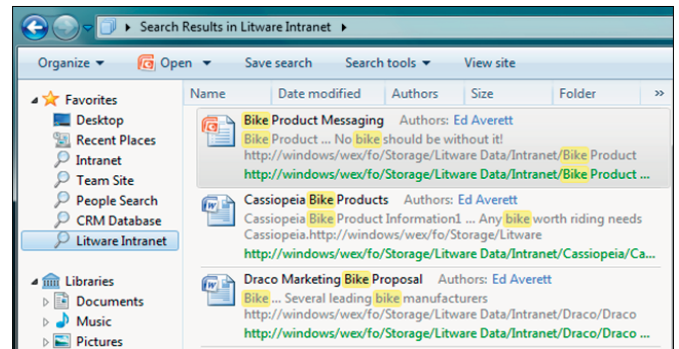
## Direct vanaf bijvoorbeeld een SharePoint-website de locatie van een bitmap binnenkrijgen

Player vinden de muziek alsof alles in één folder staat. Ook zoeken in Windows 7 gaat eenvoudiger. Windows Vista introduceerde 'Advanced Query Syntax', om bijvoorbeeld foto's van een bepaalde datum te vinden door als zoekwoord 'datetaken: 31-10-2008' mee te geven. Of muziek van een bepaalde artiest met behulp van 'artist:pink'. Veel gebruikers weten dit niet, zodat Microsoft de zoekoptie rechtsboven in Explorer heeft uitgebreid en de zoekcriteria als hints meegeeft.

Windows 7 breidt het zoeken naar bestanden verder uit met 'federated search'. Veel bestanden staan niet meer op je eigen computer, maar bijvoorbeeld binnen SharePoint, op het netwerk of internet. Je mist dan de mogelijkheid te zoeken vanuit Explorer en metadata en thumbnails te vinden. Hiervoor ondersteunt Windows 7 de bestaande standaard OpenSearch. Op basis van een RSS of ATOM feed met resultaten van een zoekopdracht toont Windows Explorer gegevens. De feed is uit te breiden met informatie over bijvoorbeeld thumbnails en metadata zoals copyright, datum,

```
System.Windows.Forms.OpenFileDialog ofd = new OpenFileDialog();
ofd.CustomPlaces.Add(@"c:\temp");
ofd.CustomPlaces.Add(@"c:\users\dennis");
ofd.ShowDialog();
```

CODEVOORBEELD 3. OPEN EEN BESTANDDIALOG MET TWEE TIJDELIJKE FAVORIETE FOLDERS




AFBEELDING 6. FEDERATED SEARCH IN WINDOWS EXPLORER

enzovoort. Diverse websites en applicaties ondersteunen deze methode al, wat direct extra potentie biedt binnen Windows 7.

Je wilt deze uitbreidingen waarschijnlijk in je eigen Windows-applicatie gaan gebruiken via de nieuwe dialoogschermen die in .NET Framework 2.0 SP1 zijn geïntroduceerd. Deze maken momenteel al gebruik van Windows Vista-features, zoals het toevoegen van favoriete locaties en het zoeken binnen folders. Een voorbeeld hiervan vind je in codevoorbeeld 3. Met Windows 7 komen hier extra uitbreidingen op. Je krijgt bijvoorbeeld de collectie van folders uit je muziekbibliotheek terug, in plaats van de huidige enkele folder in Windows Vista en voorlopers. Maar ook krijg je met de nieuwe zoekmogelijkheden direct vanaf bijvoorbeeld een SharePoint-website de locatie van een bitmap binnen om deze vervolgens in je eigen applicatie te openen en te gebruiken. Er is nog veel meer aan te passen, zoals het tonen van thumbnails van bestanden waarvoor je een eigen formaat hebt gedefinieerd. We hebben hiervoor op de PDC geen voorbeeldcode gevonden. Wel gaf men aan dat dit tot de mogelijkheden gaat behoren. Je moet jezelf afvragen of je voor dergelijke functionaliteit, in verband met performance, wel managed code wilt gebruiken.

## Conclusie

Windows 7 voegt tal van gebruiksvriendelijke features toe. Voor developers zeker interessant, omdat je hier al snel zonder al te veel moeite in je eigen applicaties gebruik van kunt maken. Daarnaast is het mogelijk Windows 7 uit te breiden door er in je applicatie rekening mee te houden, zoals aangetoond bij search. Maak je hier gebruik van, dan komt dit zeker ten goede aan de gebruikers. Zij zullen een positievere ervaring hebben met jouw applicatie. 

Dennis van der Stelt en Alex Thissen zijn als trainer en coach werkzaam bij Class-A. Beiden hebben hun eigen weblog op resp. <http://bloggingabout.net/blogs/dennis/> en <http://www.alexthissen.nl/>