

Vista Media Center Add-ins

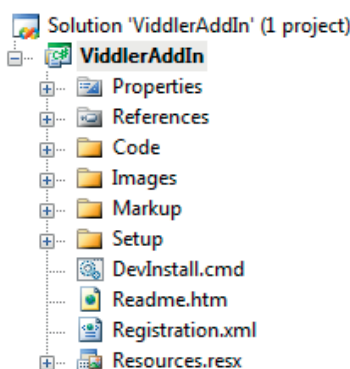
CREATIEVE VRIJHEID VOOR DE COUCH POTATO

Windows Media Center bestaat al verscheidene jaren, maar de populariteit neemt pas de laatste tijd sterk toe. Naast de standaard functionaliteit van Media Center kun je ook zelf uitbreidingen ontwikkelen. Dit was vóór het Vista-tijdperk niet eenvoudig en de integratie met Media Center verliep niet optimaal. Vista Media Center biedt echter nieuwe mogelijkheden die het gemakkelijk en vooral leuk maken om aan de slag te gaan met je eigen creatieve ideeën.

Windows Media Center is door Microsoft geïntroduceerd in 2002. In die tijd ging het om een aparte variant van het Windows XP operating system en alleen verkrijgbaar in combinatie met een Media Center-computer van een selecte groep systeembouwers. In 2005 werd het voor andere systeembouwers mogelijk computers te leveren met Windows Media Center waardoor de populariteit sterk toenam. Media Center is een applicatie voor weergave op een televisie en te bedienen met een afstandsbediening. Volgens de visie van Microsoft is Media Center het centrum van alle media in het huishouden. Dit houdt in dat je met Media Center televisie kunt kijken en opnemen, naar muziek kunt luisteren en foto's kunt bekijken. Sindsdien is het ook mogelijk uitbreidingen, zogenaamde 'add-ins', te ontwikkelen voor Media Center. Sinds Vista Media Center kun je dezelfde grafische stijl en animaties doorvoeren in je add-ins. Daarbij kun je ook gebruikmaken van de .Net programmeertalen C# of VB.Net om je add-in te bouwen.

Wat je nodig hebt

Om te beginnen heb je een willekeurige versie van Visual Studio 2005 of 2008 nodig. Verder de Windows Media Center SDK. De SDK bevat alle tools en Visual Studio-templates voor het ontwikkelen van add-ins. Hoewel je niet strikt noodzakelijk Vista Media Center hoeft te hebben, is het wel handig wanneer je de Media Center API wilt gebruiken. Die is namelijk alleen binnen Media Center aanspreekbaar. Vista Media Center wordt standaard geleverd bij Vista Home Premium en Vista Ultimate.



AFBEELDING 1. TYPISCHE ADD-IN PROJECTSTRUCTUUR

Een nieuw project starten

Je kunt twee soorten add-ins ontwikkelen voor Media Center: background add-ins en de 'gewone' add-ins. Background add-ins draaien in de achtergrond en hebben geen gebruikersinterface. We richten ons hier op de add-ins met gebruikersinterface, aangezien dit het meest is veranderd ten opzichte van vorige versies van Media Center. Ter illustratie van dit artikel zul je een add-in ontwikkelen voor het bekijken van video's die te vinden zijn op Viddler.com, een videowebsite vergelijkbaar met YouTube. Met de add-in kan je vanaf je comfortabele bank door het aanbod van Viddler bladeren. De Viddler API maakt het op het moment van schrijven nog niet mogelijk de video's af te spelen, maar dit wordt binnenkort verwacht. In dit voorbeeld maak je gebruik van een .Net wrapper voor de API die je kunt downloaden vanaf mijn weblog (zie onderaan artikel). Om te starten met een nieuwe add-in ga je in Visual Studio naar File > New > Project. Selecteer Windows Media Center Application uit de lijst met Windows Media Center project types. Media Center add-ins gaan ver in het scheiden van code van de presentatie. Dit zie je ook terug in de projectstructuur. In de standaard projectstructuur is een aparte map voor de code en een voor de markup. De code kun je schrijven in een willekeurige .Net-taal zoals je gewend bent. De magie van Vista Media Center add-ins zit echter in de markup: Media Center Markup Language of MCML.

Media Center Markup Language

MCML is een simpele variant van XAML, de markuptaal van Windows Presentation Foundation. Als je gewend bent met XAML te werken zul je MCML snel onder de knie hebben. MCML is, net als XAML, een declaratieve manier op basis van XML om de presentatielaag van je applicatie te beschrijven, zoals de lay-out, kleuren, vormen en animaties. MCML kent geen standaard controls, zoals een button of checkbox, maar geeft je de beschikking over basiselementen om zelf controls te bouwen. Dat dit niet zo moeilijk hoeft te zijn, zul je dadelijk zien. Eerst kijken we hoe MCML is gestructureerd. Elke control in MCML, zoals een pagina en een knop, noemen we een UI-element. Een UI-element heeft een bepaalde structuur, zoals te zien in codevoorbeeld 1.

In de MCML bevat de xml-node Properties de eigenschappen van je UI. In de Locals-sectie neem je alle lokale variabelen op. Vergelijk dit met

```

<UI Name="MyUI">
  <!-- Eigenschappen van de UI -->
  <Properties></Properties>

  <!-- Locale variabelen -->
  <Locals></Locals>

  <!-- Regels die binnen de UI worden afgevuurd -->
  <Rules></Rules>

  <!-- De inhoud van de UI -->
  <Content></Content>
</UI>

```

CODEVOORBEELD 1. STRUCTUUR UI-ELEMENT

de private variabelen van een class. Bij de Rules neem je regels op die worden uitgevoerd wanneer aan een bepaalde conditie is voldaan. Als laatste definieer je in de Content-sectie hoe je UI er uit ziet door andere UI-elementen aan deze sectie toe te voegen. Om verscheidene UI-elementen aan de Content-sectie toe te voegen, moet je gebruikmaken van Container-UI's, want de Content-sectie van een UI kan slechts één ander UI-element bevatten. Voorbeelden van een Container-UI zijn ColorFill, Clip en Panel.

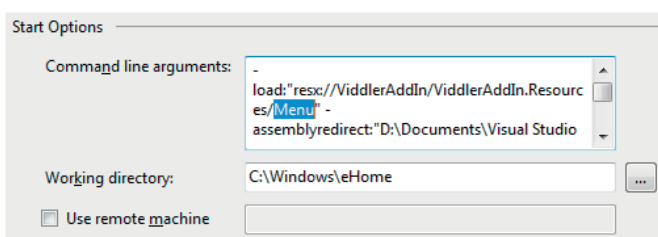
Een lijst knoppen

Als je Menu.mcml in de Markup-map opent, zie je een UI met de naam 'Default'. Je gaat deze MCML als uitgangspunt voor je add-in gebruiken. Voordat dit mogelijk is, moet je een aantal aanpassingen doen. Allereerst pas je de debug-configuratie van je project aan zodat deze UI gebruikt wordt. Open de eigenschappen van het project en ga naar de tab 'Debug'. Pas vervolgens de command line arguments aan zodat 'Menu' wordt geladen in plaats van 'Test' (zie afbeelding 2).

Verplaats de application van de Properties- naar de Locals-sectie. De reden hiervoor is dat dit de hoofd-UI is en er geen application-object meegegeven zal worden. Door de application aan de Locals-sectie toe te voegen, vertel je MCML om een nieuwe instantie van de application-class te creëren. De aangepaste Menu.mcml vind je terug in codevoorbeeld 2. De wijziging is vetgedrukt.

Nu we de aanpassingen hebben gedaan, kijken we verder naar de inhoud van deze MCML. In het voorbeeld staat onder andere de xmlns:a prefix gedefinieerd. Prefixes gebruik je om te verwijzen naar resources, zoals assemblies, afbeeldingen of andere MCML-bestanden. Helemaal bovenin een MCML-bestand in het MCML-element staan de prefixen gedefinieerd. De xmlns:a prefix wijst bijvoorbeeld naar de ViddlerAddIn namespace in de ViddlerAddIn assembly.

Deze UI bevat nu ook een Content-sectie die een ColorFill bevat. De ColorFill heeft een Repeater, die sterk lijkt op die van ASP.NET en die wordt gebruikt om verzamelingen te tonen. In dit geval staat het Source-attribuut ingesteld op [Application.MyData]. De blokhaken gebruik je in MCML om variabelen te gebruiken. Dit betekent dat de repeater de MyData property van het Application-object zal gebruiken, zoals



AFBEELDING 2. DEBUG COMMAND LINE ARGUMENTS

```

<Mcml xmlns="http://schemas.microsoft.com/2006/mcml"
  xmlns:cor="assembly://mscorlib/System"
  xmlns:a="assembly://ViddlerAddIn/ViddlerAddIn"
  xmlns:s="resx://ViddlerAddIn/ViddlerAddIn.Resources/Styles"
  xmlns:r="resx://ViddlerAddIn/ViddlerAddIn.Resources/RepeatItem">

  <UI Name="Default">
    <Properties>
      <Color Name="BackgroundColor" Color="color://s:BackgroundColor"/>
    </Properties>

    <Locals>
      <a:Application Name="Application"/>
    </Locals>

    <Content>
      <ColorFill Content="[BackgroundColor]" Layout="Center">
        <Children>

          <Repeater Source="[Application.MyData]" >
            <Layout>
              <FlowLayout Orientation="Vertical" Spacing="4,0"/>
            </Layout>
            <Content>
              <r:RepeatItem Text="[RepeatedItem!cor:String]"
                Application="[Application]" />
            </Content>
          </Repeater>
        </Children>
      </ColorFill>
    </Content>

  </UI>
</Mcml>

```

CODEVOORBEELD 2. DEFAULT UI

gedefinieerd in de Locals-sectie. De Content-sectie van de Repeater bevat een RepeatItem. Omdat RepeatItem staat gedefinieerd in RepeatItem.mcml is de prefix 'r' gebruikt om naar dit bestand te verwijzen. Wanneer je met MCML werkt is het best practice om alle MCML-bestanden in een resource file te stoppen. Vandaar dat de prefix 'r' verwijst naar een resource met de naam 'RepeatItem'. Dit heeft als voordeel dat alle MCML-bestanden tijdens het compileren worden opgenomen in de assembly en je ze niet apart hoeft mee te leveren. Afbeelding 3 toont het resultaat van het uitvoeren van de applicatie.

Afbeelding van de video

In plaats van tekst willen we een afbeelding van de video tonen. Voordat je dit kunt doen, moet je eerst zorgen dat de RepeatItem geschikt is voor het tonen van een afbeelding. Hiervoor pas je RepeatItem.mcml aan. In codevoorbeeld 3 zijn de wijzigingen vetgedrukt.

Allereerst is de Text-property naar 'Url' hernoemd. Verder is de Button in de Contentsectie met een ColorFill vervangen. De ColorFill heeft een grijze kleur en een padding van vijf pixels aan alle zijden, waardoor een rand ontstaat. Verder is het formaat van de ColorFill beperkt tot 200x150 pixels. Het CenterPointPercent geeft het middelpunt van deze UI aan en wordt uitgedrukt in de x-, y- en z-as. Dit is belangrijk als we de ColorFill bijvoorbeeld willen schalen, zoals verderop in dit artikel. De ColorFill heeft een Graphic, die een Image als content verwacht. Wij hebben slechts de URL als String. Gelukkig is MCML heel goed in het interpreteren van variabelen, zoals we in het voorbeeld zien. Door een uitroepteken achter een variabelenaam te plaatsen, geven we aan hoe de waarde geïnterpreteerd moet worden. In dit geval interpreteren we de String als Image, waardoor MCML de String gebruikt als URL voor de afbeelding.

Programmeer je eigen toekomst!

Health Software is marktleider in kennisintensieve software voor de Nederlandstalige sport- en gezondheidsbranche.

Deze lijn gaan we mondiaal doortrekken.

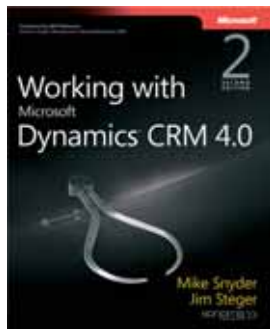
Wij zijn op zoek naar een topprogrammeur, die dit proces kan versnellen. Je houdt je bezig met object georiënteerd programmeren en bent actief betrokken bij de ontwikkeling van de diverse productlijnen. Hierdoor zul je direct een basispositie in ons team verwerven en daarmee tevens voor een deel eigenaar worden!

Ga voor meer informatie naar www.healthsoftware.nl/vacature



alle boeken van Microsoft Press zijn te verkrijgen bij:

computercollectief



13488-A3 isbn: 9780735623781

Working with Microsoft Dynamics CRM 4.0

Second Edition € 38,90

Customize Microsoft Dynamics CRM for your business - with or without programming. Configure, adapt, and extend Microsoft Dynamics CRM - with in-depth information from the experts. This guide gives you real-world guidance on how to make Microsoft Dynamics CRM work the way you do.



2135-H7 isbn: 9780735625105

MCITP Self-Paced Training Kit: Windows Server Administration

€ 54,90

Ace your preparation for the skills measured by MCITP Exam 70-646 - and on the job. Work at your own pace through a series of lessons and reviews that fully cover each exam objective. Then, reinforce and apply what you've learned through real-world case scenarios and practice exercises.



13873-A6 isbn: 9780735622791

Windows PowerShell Scripting Guide

€ 38,90

Your hands-on guide to automating administration using Windows PowerShell. Discover how to automate the day-to-day management of Windows Server 2008 and Windows Vista using Windows PowerShell, the scripting language for system administrators. Includes a companion CD with 3090+ sample scripts and more.



13491-A8 isbn: 9780735623613

Windows Server 2008 Resource Kit

€ 189,90

Scripts, eBooks, Digital Resources on CD; Internet Information Services 7.0 Resource Kit; Windows Administration Resource Kit; Windows PowerShell Scripting Guide; Windows Server 2008 Active Directory Resource Kit; Windows Server 2008 Security Resource Kit; Windows Server 2008 Networking and Network Access Protection.

computercollectief b.v.

e-mail: info@comcol.nl

tel: 020 6223573

Prijzen incl. btw en onder voorbehoud



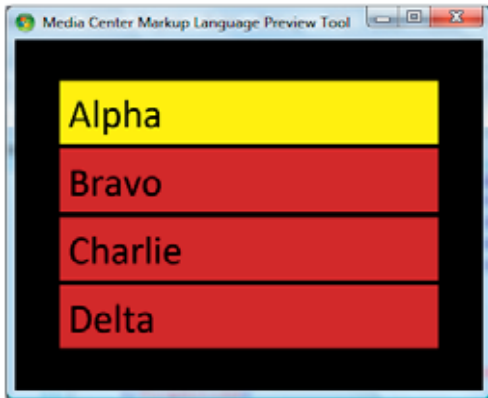
comcol.nl

Alle artikelen zijn te bestellen via **comcol.nl**

maar ook te kopen in onze **winkel:**

Amstel 312 (tegenover Carré)

1017 AP Amsterdam



AFBEELDING 3. VIER KNOPPEN

Nu is het moment gekomen waarop je gebruik gaat maken van de Viddler.Net API. In de meegeleverde uitleg bij de Viddler.Net API staat beschreven hoe je een Viddler API Key moet aanvragen om de API te kunnen gebruiken. Verder staat er ook in welke wijzigingen je aan de configuratie van het project moet aanbrengen om de API Key te gebruiken. Na het doorlopen van deze stappen kun je de Viddler.Net API gebruiken door een referentie naar `Avanade.ViddlerNet.dll` toe te voegen aan het project. Vervang de statische String array met iets interessants, namelijk een array van video's. Verwijder hiervoor de `MyData` property uit de Application class en vervang deze door het stukje code uit codevoorbeeld 4.

De code in codevoorbeeld 4 haalt maximaal twintig video's op van de gebruiker 'avatube' en retourneert de video's als een array van het type `Avanade.ViddlerNet.Video`. De volgende stap is het aanpassen van de source van de Repeater zodat de nieuwe Videos property wordt gebruikt. Omdat de Repeater nu over Video-objecten in plaats van Strings itereert moet je ook de interpretatie aanpassen. Aangezien de Video-class in de Viddler.Net assembly zit moet je een verwijzing naar deze assembly opnemen. Het resultaat is weergegeven in codevoorbeeld 5. De aanpassingen zijn vetgedrukt.

Reageren op input

We hebben nu een lijst met foto's van een aantal video's. De volgende stap is het mogelijk maken om een video in de lijst te selecteren. Je hoeft hiervoor slechts aan te geven dat er interactie mogelijk is met het RepeatItem. Dit doe je door de regel in codevoorbeeld 6 toe te voegen aan de Rules-sectie.

Dit is een zogenaamde Default-rule. Default-rules worden altijd één keer aangeroepen wanneer een UI geïnstantieerd wordt. In dit geval zorg

```
<UI Name="RepeatItem">
  <Properties>
    <a:Application Name="Application" Application="$Required"/>
    <cor:String Name="Url" String="$Required"/>
  </Properties>

  <!-- ... -->

  <Content>
    <ColorFill Name="Background" Content="Gray" Padding="5,5,5,5"
      MinimumSize="200,150" MaximumSize="200,150" CenterPointPer-
      cent=".5,.5,.5">
      <Children>
        <Graphic Name="Thumbnail" Content="[Url!Image]"/>
      </Children>
    </ColorFill>
  </Content>
</UI>
```

CODEVOORBEELD 3. REPEATITEM MET AFBEELDING

```
public Avanade.ViddlerNet.Video[] Videos
{
  get
  {
    Avanade.ViddlerNet.ViddlerNet viddlerNet = new Avanade.Vid-
    dlerNet.ViddlerNet();
    Avanade.ViddlerNet.Video[] videos = viddlerNet.
    GetVideosByUser("avatube").Videos;
    return videos;
  }
}
```

CODEVOORBEELD 4. PROPERTY VOOR OPHALEN VAN VIDEO'S

```
<Mcml xmlns="http://schemas.microsoft.com/2006/mcml"
  xmlns:cor="assembly://mscorlib/System"
  xmlns:a="assembly://ViddlerAddIn/ViddlerAddIn"
  xmlns:s="resx://ViddlerAddIn/ViddlerAddIn.Resources/Styles"
  xmlns:r="resx://ViddlerAddIn/ViddlerAddIn.Resources/Repea-
  tItem"
  xmlns:v="assembly://ViddlerNet/ViddlerNet">

  <UI Name="Default">
    <!-- ... -->

    <Content>
      <ColorFill Content="[BackgroundColor]" Layout="Center">
        <Children>

          <Repeater Source="[Application.Videos]">
            <!-- ... -->
            <Content>
              <r:RepeatItem Url="[RepeatedItem!v:Video.Thumbnail-
              Url]"
                Application="[Application]"/>
            </Content>

          </Repeater>
        </Children>
      </ColorFill>
    </Content>

  </UI>
</Mcml>
```

CODEVOORBEELD 5. AANGEPASTE REPEATER

je ervoor dat de property `Input.KeyInteractive` de waarde 'true' krijgt toegekend. Deze ingebouwde property zorgt ervoor dat deze UI is te selecteren met muis, toetsenbord en afstandsbediening. Hoewel het nu mogelijk is een RepeatItem te selecteren, kun je nog niet zien welk item op een bepaald moment is geselecteerd. Hiervoor voeg je twee nieuwe rules toe. De eerste zorgt voor het oplichten van een geselecteerd item en de tweede rule voor het omgekeerde effect wanneer het item niet meer geselecteerd is. Deze twee rules zie je in codevoorbeeld 7.

In dit geval maak je gebruik van een Condition-rule, die wordt uitgevoerd zolang aan een bepaalde voorwaarde wordt voldaan. In dit geval het controleren van de property `Input.KeyFocus`. Deze geeft aan of de UI focus heeft of niet. Vervolgens worden Actions uitgevoerd wanneer aan de voorwaarde wordt voldaan. In dit geval krijgt de ColorFill met de naam Background een andere kleur en wordt de schaal aangepast, zodat de UI bij selectie een klein beetje groter is dan de rest. De tweede Condition-rule maakt deze wijziging weer ongedaan op het moment dat het item geen focus heeft. Voer het project uit en merk op dat het nu mogelijk is om met de pijltjestoetsen of de muis een video aan te wijzen.

Van links naar rechts

Het zal je waarschijnlijk zijn opgevallen dat de lijst met video's te lang is om in zijn geheel te tonen. Je zult een manier moeten bedenken om door

de lijst te scrollen. MCML heeft hier een handig hulpmiddel voor: de Scroller. De Scroller is een Container-UI, net als ColorFill. De Scroller zorgt er voor dat je door zijn child-elementen kunt scrollen wanneer ze niet allemaal op het scherm passen. Een Scroller is heel eenvoudig toe te passen. Om dit aan te tonen vervang je de ColorFill in Menu.mcml met een Scroller. In dit geval geef je de Scroller twee attributen mee. Het eerste attribuut met de naam Orientation bepaalt of de Scroller horizontaal of verticaal moet scrollen. Omdat de lijst met video's verticaal wordt getoond, zet je de Orientation van de Scroller ook op verticaal. Het tweede attribuut voegt een vervaging toe aan de uiteinden van de Scroller. Dit geeft de eindgebruiker de indruk dat er meer video's zijn. De UI met Scroller ziet er uit als in codevoorbeeld 8.

Dat kan vloeiender

Je kunt nu scrollen door de lijst met video's en een video selecteren. Alleen ziet het er allemaal nog niet zo vloeiend uit als de rest van Media Center. Je laat het scrollen wat vloeiender verlopen door een animatie te specificeren. Deze animatie pas je toe op de Repeater. De Repeater met animatie vind je terug in codevoorbeeld 9.

Iedere Container-UI kan animaties bevatten. Animaties specificer je in een Animations-sectie van de UI. In het voorbeeld hebben we een animatie van het type 'Move' gespecificeerd. Dit type animatie zorgt voor een vloeiende beweging van een UI tussen twee of meer punten. Als je het project uitvoert, zul je zien dat het scrollen door de lijst vloeiender gaat.

Conclusie

De Vista Media Center SDK betekent een grote stap voorwaarts in het ontwikkelen van add-ins voor Media Center. Je kunt nu mogelijk dezelfde look-and-feel hanteren in je zelfontwikkelde uitbreidingen. Toch

```
<Default Target=" [Input.KeyInteractive]" Value="True"/>
```

CODEVOORBEELD 6. INTERACTIE MOGELIJK MAKEN

```
<Condition Source=" [Input.KeyFocus]" SourceValue="true">
  <Actions>
    <Set Target=" [Background.Content]" Value="White"/>
    <Set Target=" [Background.Scale]" Value="1.05,1.05,1.05"/>
  </Actions>
</Condition>

<Condition Source=" [Input.KeyFocus]" SourceValue="false">
  <Actions>
    <Set Target=" [Background.Content]" Value="Gray"/>
    <Set Target=" [Background.Scale]" Value="1,1,1"/>
  </Actions>
</Condition>
```

CODEVOORBEELD 7. GESELECTEERDE VIDEO OPLICHTEN

```
<UI Name="Default">
  <!-- ... -->

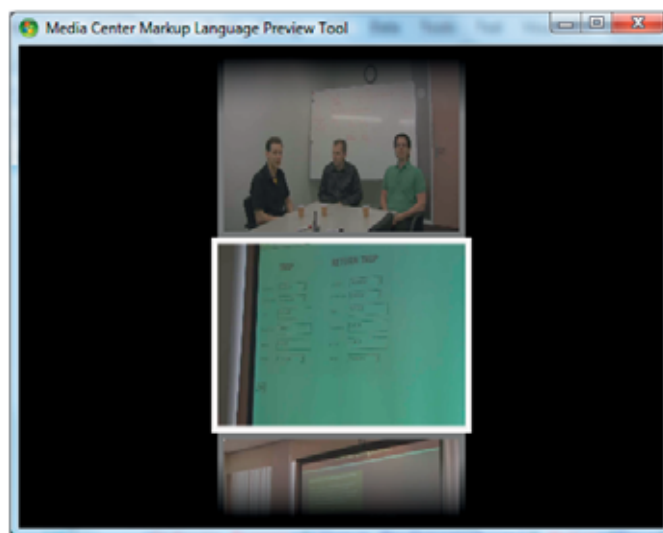
  <Content>
    <Scroller Orientation="Vertical" FadeSize="50">
      <Children>
        <Repeater Source=" [Application.Videos]" >
          <!-- ... -->
        </Repeater>
      </Children>
    </Scroller>
  </Content>

</UI>
```

CODEVOORBEELD 8. SCROLLER

```
<Repeater Source=" [Application.Videos]" >
  <!-- ... -->
  <Animations>
    <Animation Type="Move">
      <Keyframes>
        <PositionKeyframe Time="0.00" RelativeTo="Current"/>
        <PositionKeyframe Time="0.25" RelativeTo="Final"/>
      </Keyframes>
    </Animation>
  </Animations>
</Repeater>
```

CODEVOORBEELD 9. SCROLL-ANIMATIE



AFBEELDING 4. EINDRESULTAAT

kleeft er nog wel een aantal beperkingen aan het ontwikkelen van add-ins. Misschien wel de grootste is het ontbreken van standaard controls, zoals die in de rest van Media Center worden gebruikt. Microsoft heeft hier bewust voor gekozen omdat commerciële bedrijven vaak juist een afwijkende stijl willen hanteren om hun product herkenbaar te maken. Hierdoor blijft het nog wat werk om exact dezelfde stijl als Media Center te krijgen. Een initiatief om een bibliotheek te ontwikkelen met dit soort controls is helaas gestrand vanwege gebrek aan animo. Er valt nog een heleboel te schrijven over het bouwen van Media Center add-ins. Ga zelf maar eens op onderzoek uit! Er is een aantal goede resources op het Internet die je verder op weg kunnen helpen. De beste daarvan vind je in de links onderaan dit artikel. Hopelijk heb je na het lezen van dit artikel zin gekregen om je eigen Vista Media Center add-in te ontwikkelen. Zodat we in de toekomst nóg meer plezier kunnen hebben van ons Media Center. **.net**

Links

Product team blog en links naar de SDK
<http://blog.mediacentersandbox.com/default.aspx>

The Digital Lifestyle Developer Blog
<http://thedigitallifestyle.com/cs/blogs/developer/default.aspx>

Viddler.com
<http://www.viddler.com>

Sander Schutten's weblog
<http://www.afanaat.nl>

.....
Sander Schutten is Senior Programmer bij Avenade.