

Gave gadgets bouw je zelf

GADGETS VOOR DE WINDOWS SIDEBAR

Nieuw in Windows Vista is de Windows Sidebar waarop allerlei gadgets actief zijn die informatie weergeven of een service aanbieden. Standaard wordt er een aantal handige gadgets meegeleverd maar het is vrij eenvoudig om zelf een gadget te bouwen.

Aan de ene kant hebben we te maken met een overvloed aan informatie, aan de andere kant zijn er zaken die je direct zichtbaar wilt hebben. Denk bijvoorbeeld aan het benzinemetertje op je dashboard, af en toe werp je er een blik op en weet je of je moet tanken of nog even door kunt rijden. Bij Windows hebben we eigenlijk alleen rechtsonder een klein hoekje voor statusinformatie. Dankzij de alsmar hoger wordende beeldschermresolutie, de komst van 16x9 en 16x10 schermen en multi-monitor systemen, hebben we tegenwoordig veel ruimte op het scherm vrij voor een echt dashboard. In Windows Vista introduceert Microsoft de Windows Sidebar. De Windows Sidebar toont op een vast gedeelte van het beeldscherm actuele informatie die voor de gebruiker van belang is. De Windows Sidebar is het directe gevolg van een onderzoeksproject van Microsoft Research. Er werd naar een manier gezocht om informatie op een efficiënte manier te pushen –ten tijde van het onderzoek sprak men nog over pointcast en channels. In de zomer van 2000 heeft MS Research een prototype gebouwd, genaamd Sideshow. In een rap tempo verschenen er allerlei handige add-ins voor de Sideshow terwijl het slechts een prototype was! Ook andere dashboards zoals de Desktop Sidebar, het Apple Dashboard en de Yahoo Widget Engine (ook bekend als Konfabulator) genieten grote populariteit bij ontwikkelaars en gebruikers. In plaats van widget, webparts, pannels, add-ins of iets dergelijks spreken we bij Microsoft over gadgets. In dit artikel behandel ik alleen gadgets voor de Windows Sidebar.

Een gadget is een miniapplicatie die informatie toont of een bepaalde taak uitvoert. De bekendste toepassingen zijn onder andere een klokje, rss-reader, mediaplayer, weersverwachting, ver-

keersinformatie, status van een bestelling, enzovoort. Naast gadgets die vooral persoonlijk nuttig zijn zoals nieuws, agenda, sportuitslagen, zijn er ook veel toepassingen denkbaar in de zakelijke sfeer. Het direct zichtbaar hebben van bijvoorbeeld de status van het netwerk of e-mailserver, bezettingsgraad, KPI's van de afdeling, beschikbaarheid van vergaderruimtes, tracking van een pakketje, aantal inschrijvingen voor trainingen/evenementen, enzovoort. Een belangrijk principe van een gadget is dat hij weinig code en schermruimte vraagt. Gadgets zijn bedoeld als miniapplicaties, één taak per gadget. Dus: "keep it simple, less is more". Hoe aantrekkelijk misschien ook in de ogen van sommige marketeers, gadgets zijn niet bedoeld voor advertenties.

Hoe bouw je een gadget

Een gadget is gebaseerd op webtechnologie, het is dus primair een DHTML-pagina met een bijbehorend script. Een gadget heeft een host-omgeving nodig waarvan er drie zijn: Als eerste is er de Windows Sidebar; zie afbeelding 1. Gadgets kunnen op de Sidebar geplaatst worden of 'floating' op de desktop – zoals de calculator. Voor de gadgets draait de host-omgeving sidebar.exe. Een gadget is eigenlijk een 'child window' van sidebar.exe. Je zult zien dat het proces sidebar.exe tweemaal draait. Het tweede sidebar.exe-proces is een sandbox voor gadgets. Naast de Windows Sidebar zijn er al geruime tijd op het internet twee host-omgevingen voor gadgets beschikbaar. Dit zijn Windows Live www.live.com en www.start.com. Als laatste is er ook nog de Windows SideShow. Een nieuwe generatie laptops is voorzien van een klein extra auxiliary display; zie afbeelding 2. Informatie op het auxiliary display is ook zichtbaar als de laptop is dichtgeklapt en zelfs als de laptop uit staat. Via het auxiliary display kun je snel je agenda inzien, een telefoonnummer opzoeken of e-mail lezen, allemaal als je laptop uit staat.

De overeenkomst tussen Windows Sidebar en Windows Live en Start is dat ze alle drie op DHTML gebaseerd zijn. Het voordeel is dat de presentatielaag en achterliggende logica van de gadgets gelijk blijft. Alleen een paar kleine aanpassingen zijn nodig en de manier van verpakken en installeren is anders. Gadgets voor de Windows SideShow is een ander verhaal, omdat de onderliggende technologie voor de Sideshow afwijkt. In feite is het een los device met een eigen OS (tiny CLR) en is het verbonden



Afbeelding 1. Gadgets in de Windows Sidebar en 'floating' op de desktop.

Dit artikel is gebaseerd op de daily build 5376 van Windows Vista (Nederlands). Elke dag wordt er een nieuwe build van Windows Vista gemaakt, ook wel 'self host build' genoemd. Sinds de PDC 05 en zelfs CTP build van februari 2006 is er heel veel veranderd. Veel informatie is gebaseerd op de CTP build van februari 2006 en is helaas verouderd. De planning is om de informatie te updaten voor de bètaversie 2. Het is goed mogelijk dat er in de tussentijd nog kleine aanpassingen komen.

aan de accu van de laptop en via een interface bereikbaar. Zie 'The 10 Show' voor een korte impressie <http://on10.net/The-Show/2890/>

Hello World

Tijd om te beginnen met een simpel voorbeeld. Gadgets worden lokaal geïnstalleerd in de volgende directory: %userprofile%\AppData\Local\Microsoft\Windows Sidebar\Gadgets, waarbij opgemerkt moet worden dat de folder \Documents and Settings bij Windows Vista vervangen is door \user. Maak een nieuwe folder in \Gadgets met de naam PowerMonitor.gadget. In deze folder plaatsen we dadelijk alle bestanden voor onze gadget. Als we later de gadget willen distribueren, kunnen we ervoor kiezen om alle bestanden in een Windows Cabinet file (.cab) of een zip-bestand te plaatsen en deze dan ook de gadget-extensie te geven. Aan de gadget-extensie herkent de Sidebar de gadget en zal deze installeren. Het voordeel van Windows Cabinet is dat we code-signing kunnen toepassen. Voor het gemak kiezen we er nu voor alle bestanden los in de directory te plaatsen. In de folder voor je gadget creëer je als eerste een manuscript genaamd gadget.xml; zie codevoorbeeld 1. In het manuscript staat informatie over de gadget met als belangrijkste de verwijzing naar het bestand met de code, in dit geval PowerMonitor.html. Ook dit bestand maken we aan zoals aangegeven in codevoorbeeld 2. De maximale breedte van de gadget is 130px, daar dien je dus binnen te blijven. In principe is het 'Hello World'-voorbeeld nu klaar om geïnstalleerd te worden. Mocht je de Sidebar nog niet gestart hebben, dan vind je die bij Bureau-accessoires. Door op het plusje te klikken (tegenwoordig boven in de Sidebar) kun je een gadget toevoegen aan de Sidebar. Als het goed is staat onze 'Power Monitor'-gadget er ook tussen. Als je gadget er wel tussen staat maar een grijze kleur heeft, is er iets mis met het manuscript. Corrigeer het probleem en geef je directory van je gadget een andere naam. Het aanpassen en opnieuw opslaan van het manuscript werkt niet. Pas als er een aanpassing gemaakt is in de directorystructuur waar alle gadgets zijn ondergebracht, zal de gadget-installer de nieuwe gadget.xml inlezen. Een snellere methode is het verwijderen van het bestand settings.ini in de userfolder



Afbeelding 2. Auxiliary display met Windows Side Show op nieuwe generatie laptops

```
<?xml version="1.0" encoding="utf-8" ?>
<gadget>
  <name>Power Monitor</name>
  <namespace>microsoft.windows</namespace>
  <version>1.0.0.0</version>
  <author name="Robert Fransen">
    <logo src="logo.png"/>
    <info url="www.microsoft.nl/gadget"/>
  </author>
  <copyright>&#169;2006 .NET Magazine</copyright>
  <description>Statusmonitor voor de laptopbatterij</description>
  <icons>
    <icon height="48" width="48" src="icon.png"/>
  </icons>
  <hosts>
    <host name="sidebar">
      <base type="HTML" apiVersion="1.0.0" src="PowerMonitor.html"/>
      <permissions>full</permissions>
      <platform minPlatformVersion="0.3" />
    </host>
  </hosts>
</gadget>
```

Codevoorbeeld 1. Manuscript voor de gadget

```
<html>
<head>
</head>
<body style="height:62px; width:130px;">
Hello World
</body>
</html>
```

Codevoorbeeld 2. Hello World gadget.

onder Window Sidebar, waardoor alle gadget manifest-bestanden opnieuw ingelezen worden. Dit is echter niet ideaal want je raakt ook andere statusinformatie over de gadget kwijt.

Gadget API's

Door een kleine aanpassing kun je de gadget wat nuttigs laten weergeven. Het gadget-platform biedt een rijk objectmodel voor interactie met Windows. Er zijn APIs voor het lezen en schrijven van bestanden, voor het opvragen van netwerkinformatie, CPU-status en batterijstatus. Die laatste gaan we gebruiken. Het object *System.PowerStatus* biedt een property genaamd *batteryLifePercent*. Deze geeft een integer terug met het percentage van het resterende vermogen van de batterij. Zie codevoorbeeld 3 voor de toepassing van het object *System.PowerStatus*. Met behulp van een simpele functie *getStatusInfo* kun je de waarde ophalen en weergeven in de gadget. Wanneer je geen accu in je systeem hebt, kun je *System.PowerStatus.batteryLifePercent* vervangen door *System.Machine.CPUUsagePercentage*. Je krijgt dan het gebruik van processor-cycles te zien. Wanneer je een aanpassing aan de HTML-code hebt gemaakt, moet je de gadget opnieuw toevoegen aan de Sidebar. De Sidebar ondersteunt multiple instances, wat bij het testen erg gemakkelijk kan zijn.

Events

Zoals in afbeelding 1 te zien is kunnen gadgets op de Sidebar worden weergegeven; de status is dan *docked*. Daarnaast kunnen ze ook vrij 'floating' op de desktop geplaatst worden; de status is dan *undocked*. Wanneer een gebruiker de gadget van de Sidebar naar de desktop sleept of vice versa, kun je het dock/undock-event afvangen. Het aardige hiervan is dat je bijvoorbeeld de afmeting van de gadget groter kunt maken als deze op de desktop geplaatst wordt. In codevoorbeeld 4 zie je de code voor het afvangen van het undock-event waarbij via *System.Gadget.onUndock* naar een functie wordt verwezen. Het fraaie is dat we alle veranderingen (afmeting aanpassen, lettertype groter maken en een ander background image selecteren) als één actie kun-

```

<html>
<head>
<script type="text/javascript">
var timerId;
var refreshRate = 250; // milliseconden

function getStatusInfo()
{
    batteryLifePercent.innerHTML =
    System.PowerStatus.batteryLifePercent + "%";
    timerId = setTimeout("getStatusInfo()", refreshRate);
}
</ script>
</head>

<body onload="getStatusInfo()"
style="width:130px; height:60px;
font-family:Tahoma; font-size: 7pt">
<spanid="titleSpan"><strong>Powermonitor</strong></span><br />
Vermogen batterij: <span id="batteryLifePercent">xx%</span>
</body>
</html>

```

Codevoorbeeld 3.

```

System.Gadget.onUndock = undockedState;
System.Gadget.onDock = dockedState;

function undockedState()
{
System.Gadget.beginTransition();
    with(document.body.style)
        width=200,
        height=100,
        fontSize="14px",
        backgroundImage="url('background_desktop.png')";
    with(titleSpan.style)
        fontSize="16px";
System.Gadget.endTransition(System.Gadget.TransitionType.none, 0);
}

function dockedState()
{
...
}

```

Codevoorbeeld 4. PowerMonitor.js

nen laten plaatsvinden. Met de *System.Gadget.beginTransition()*-method kun je aangeven dat de volgende scriptregels een aantal visuele effecten tot gevolg hebben. Met *System.Gadget.endTransition* geef je het eind van een reeks visuele veranderingen aan en dat er van de oude view state naar de nieuwe wordt overgegaan. Bij de *endTransition*-method kun je trouwens een gewenst transitietype aangeven en het aantal seconden voor deze transitie.

Settingsmenu

Standaard is de mogelijkheid aanwezig een apart scherm te definiëren voor instellingen. Op de powermonitor.html maak je door middel van *System.Gadget.settingsUI* = "Settings.html" een verwijzing naar de pagina voor het aanpassen van instellingen, in dit geval settings.html. Wanneer je nu op de gadget op de rechtermuisknop klikt, komt de settings-dialoogbox te voorschijn. Op deze pagina kun je de gebruiker allerlei instellingen laten veranderen. Je dient in settings.html het event *System.Gadget.onSettingsClosing* af te vangen. Zie codevoorbeeld 5 en 6. Dit event vindt plaats als de gebruiker op 'ok' of 'cancel' heeft geklikt. Even afvragen wat er precies gebeurd is en vervolgens wel of niet de aangepaste instellingen wegschrijven met behulp van *System.Gadget.Settings.write()*. Op de hoofdpagina van de

```

<html>
<head>
<script type="text/javascript" src="settings.js"></script>
<script type="text/javascript">
function LoadSettings()
{
    System.Gadget.onSettingsClosing = SettingsClosing;
    document.all.refreshRate.value = GetRefreshRate();
}
</script>

</head>
<body style="width: 130; height:90px" onload="LoadSettings()">
<div style="font-size: 10px">
    Refresh rate:<br />
    <input name="refreshRate" style="width:40%;" type="text" /><br />
    Geef een waarde tussen 50 en 1000 milliseconden.
</div>
</body>
</html>

```

Codevoorbeeld 5. Settings.html

```

function SettingsClosing(event)
{
    if (event.closeAction == event.Action.commit)
    {
        SaveSettings();
    }
    else if(event.closeAction == event.Action.cancel)
    {
    }
}

function SaveSettings()
{
    try
    {
        var num = Number(document.all.refreshRateInput.value);
        if (num >= 50 && num <= 1000)
        {
            SetRefreshRate(num);
        }
    }
    catch (e)
    {
        // Doe niets, geen geldige input.
    }
}

function SetRefreshRate(rate)
{
    System.Gadget.Settings.write("RefreshRate", rate);
}

function GetRefreshRate()
{
    var refreshRate
    refreshRate = System.Gadget.Settings.read("RefreshRate");
    if (refreshRate == "")
    {
        refreshRate = 250; // milliseconden
        System.Gadget.Settings.write("RefreshRate", refreshRate);
    }
    return refreshRate;
}

```

Codevoorbeeld 6. Settings.js

```

<script type="text/javascript" src="settings.js"></script>
<script type="text/javascript">
var refreshRate = 250; // milliseconden

System.Gadget.settingsUI = "Settings.html";
System.Gadget.onSettingsClosed = SettingsUpdated;

function SettingsUpdated()
{
refreshRate = GetRefreshRate();
}

function openFlyout()
{
System.Gadget.Flyout.file = "Flyout.html";
System.Gadget.Flyout.show = true;
}
</script>
<body onmousedown="openFlyout()">
...
</body>

```

Codevoorbeeld 7. Powermonitor.html met verwijzing naar de flyout- en settings-pagina

gadget PowerMonitor.html vang je vervolgens het event *System.Gadget.onSettingsClosed* op en daarna met de functie *System.Gadget.Settings.read()* kun je aangepaste waarde weer ophalen; zie codevoorbeeld 7. Waar je wel rekening mee moet houden is dat deze manier van instellingen opslaan, geldig is zolang je gadget draait. Met een cookie kun je er bijvoorbeeld voor zorgen dat de instellingen langer bewaard blijven. Op www.microsoft.nl/netmagazine13 is de volledige code voor de Powermonitor-gadget te downloaden.

Flyout

Een breedte van 130 pixels is natuurlijk niet veel. Het is denkbaar dat je op een bepaald ogenblik meer informatie wilt laten zien. Stel dat je bijvoorbeeld een gadget met verkeersinformatie hebt. Op het moment dat de gebruiker op een wegnummer klikt, wil je dat er een kaartje wordt getoond van de desbetreffende file. Dit is mogelijk met behulp van een flyout-venster. De flyout zit verbonden aan de gadget en mag meer dan 130px breed zijn; zie afbeelding 3. In codevoorbeeld 7 is te zien hoe *onmousedown* aan de functie *openFlyout()* is gekoppeld die op zijn beurt de *flyout.html*-pagina zichtbaar maakt.



Afbeelding 3. Flyout-venster

Verschillende talen

Tot nu toe hebben we alle bestanden in een folder geplaatst, maar het is beter om afbeeldingen, javascript, enzovoort netjes in een eigen folder te plaatsen. Om je gadget verschillende talen te laten ondersteunen is dit ook de geëigende weg. Voor elke Windows locale die je ondersteunt, creëer je in de gadget-folder een eigen folder per locale. Voor Amerika is dat `\en-US`, voor Nederland is dat `\nl-NL` en voor België is dat `\nl-BE`. In de desbetreffende directory plaats je het manifest voor die combinatie taal/land. Gemeenschappelijke bestanden zoals `icon.png` kun je een folder hoger plaatsen.

Beheer

Het is mogelijk via een policy een disable van de Sidebar af te dwingen; hiervoor is regkey `HKCU\Software\Policies\Microsoft\Sidebar` beschikbaar. Standaard is de Sidebar enabled maar je kunt ook opgeven of de Sidebar op gebruiker- of op machineniveau gebruikt kan worden.

Tot slot

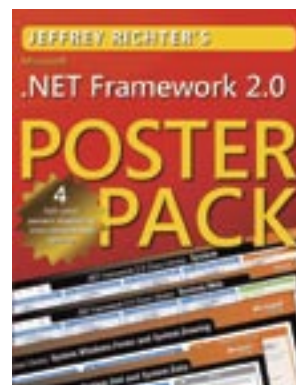
Er is voor gekozen voor DHTML (inclusief ondersteuning voor AJAX) als technologie voor het bouwen van gadgets. Eerder was er sprake van dat je gebruik kon maken van WPF (Windows Presentation Foundation). Daar is volledig van afgestapt. Wel is het mogelijk een WPF/E-pagina te embedden in de HTML-code. Het nadeel hiervan is dat je vanuit WPF/E geen gebruik kunt maken van het objectmodel van de Sidebar. In Windows wordt er standaard een aantal gadgets meegeleverd. Deze pre-installed gadgets zijn te vinden in de directory `\Program Files\Windows Sidebar\Gadgets`. Het is zeker de moeite waard de code daarvan eens te bekijken. Gadgets bieden interessante mogelijkheden voor contentproviders, maar het is ook uitermate geschikt voor een Mashup. Een Mashup of een gave gadget schrijf je natuurlijk zelf.

Robert Fransen is freelance consultant en parttime werkzaam bij Microsoft Nederland in het webteam, verantwoordelijk voor onder andere de MSDN-website. Zijn e-mailadres is robert.fransen@rcenter.nl

Referenties:

www.microsoftgadgets.com
www.microsoft.nl/gadget
www.live.com
www.start.com/developer

(advertentie Microsoft Press)



**Microsoft .NET Framework 2.0
 Poster Pack**
 ISBN: 0-7356-2317-1
 Auteur: Jeffrey Richter
 (Wintellect)