



Windows Vista - Gadgets

Minianwendungen („Gadgets“) für die Windows Vista Sidebar entwickeln

Christian Haberl

Nachdem ich im Büro recht gute PC Lautsprecher, aber kein Radio habe, höre ich gerne Internet Radio. Auch bin ich oft in Amerika und höre dann gerne österreichische Radiosender von dort aus.

Mein Musikgeschmack ist ziemlich vielfältig, ich höre gerne Radio Stephansdom, gelegentlich Ö1, manchmal FM4, und auch oft Radio Wien. Ö3 höre ich selten bis gar nicht bzw. nur beim Autofahren.

Als ich davon hörte, dass es ein Ö3-Windows Sidebar Gadget gibt, wollte ich mir dieses dennoch herunter laden, weil ich wissen wollte, was dahinter steckt, Windows Sidebar Gadgets sollen ja so schön und relativ einfach zu entwickeln sein. Leider war das Sidebar Gadget damals vorübergehend nicht verfügbar, und ein paar Webcasts und Blog-Artikel später beschloss ich einfach ein eigenes Sidebar Gadget für österreichische Radiosender zu bauen, zumal mir Ö3 alleine ohnehin nicht reichen würde.

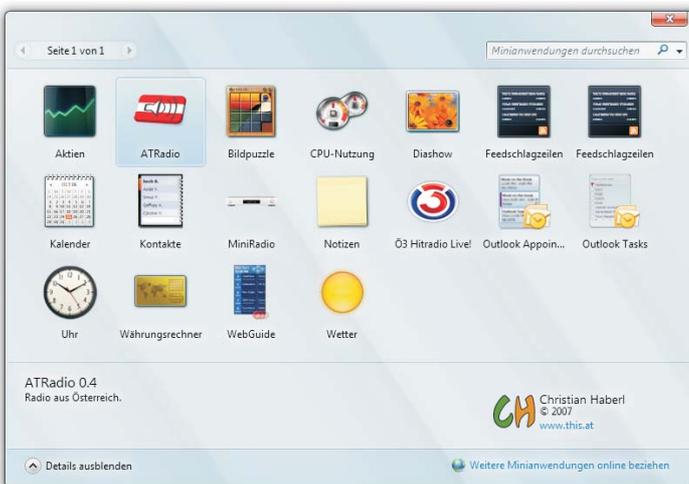
Ein Vista Gadget, auf Deutsch auch „Minianwendung“ bezeichnet, besteht im Wesentlichen aus einigen Files die im Verzeichnis

C:\Users\%USERNAME%\AppData\Local\Microsoft\Windows\Sidebar\Gadgets abgelegt sind. Das Unterverzeichnis ATRadio07.gadget enthält diese Files:

gadget.xml, atradio.html, atradio.css, atradio.js, flyout.html, flyout.css, flyout.js, settings.html, atradio.png, glass130.png, glass130c.png, logo.png



Wenn das Manifest File vorhanden ist, wird das Gadget auch automatisch im Vista Gadget Auswahlfenster angezeigt:



Das File gadget.xml, die Manifest-Datei, enthält alle Definitionen für das Gadget:

```
gadget.xml
<?xml version="1.0" encoding="utf-8"?>
<gadget>
  <name>ATRadio</name>
  <namespace>this.at.radio</namespace>
  <version>0.4</version>
  <author name="Christian Haberl">
    <info url="http://www.this.at" text="www.this.at" />
    <logo src="logo.png" />
  </author>
  <copyright>© 2007</copyright>
  <description>Radio aus Österreich.</description>
  <icons>
    <icon height="64" width="64" src="atradio.png" />
  </icons>
  <hosts>
    <host name="sidebar">
      <base type="HTML" apiVersion="1.0.0" src="atradio.html" />
      <permissions>full</permissions>
      <platform minPlatformVersion="1.0" />
      <defaultImage src="atradio.png" />
    </host>
  </hosts>
</gadget>
```

atradio.html ist der Hauptteil des Gadgets, den Javascript Code habe ich in atradio.js ausgelagert, die Styles in atradio.css.

Der Aufbau ist recht einfach: Es gibt einen Button der gleichzeitig den Radiosender anzeigt und das Flyout öffnet um den Sender zu wechseln. Es gibt ein Windows Media Player Objekt und einen Start und einen Stop Button.

Im Wesentlichen wird das Windows Media Player Objekt mittels Javascript angesteuert, um die Radiostreams abzuspielen.

Wenn ein Stream gespielt wird, zeigt das Media Center Objekt eine Visualisierung, sonst wird der Status angezeigt, etwa „Buffering...“ oder „Error“.

Error bedeutet übrigens so gut wie immer, dass der Streaming Server überlastet ist, was bei Ö3 und Radio Wien an einem Arbeitstag unter Tags recht oft vorkommt, wenigstens in letzter Zeit.

```
atradio.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>ATRadio</title>
    <link rel="stylesheet" type="text/css" href="atradio.css">
    <script src="atradio.js" language="javascript" type="text/javascript"></script>
  </head>
  <body id="gadgetContent" onload="SetupGadget();" >
    <input type="button" value="select station" id="stationname" onclick="toggleFlyout()"><br>
    <div id="whichState">...</div>
    <div id="mplayerdiv">
      <object id="Player" classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6">
        <param name="autoStart" value="False">
        <param name="SendPlayStateChangeEvents" value="True">
        <param name="SendErrorEvents" value="True">
        <param name="SendOpenStateChangeEvents" value="True">
        <param name="TransparentAtStart" value="True">
        <param name="uimode" value="none">
      </object>
    </div>
    <br>
    <input type="button" name="BtnPlay" value="play" onclick="StartMeUp()" id="BtnPlay">
    <input type="button" name="BtnStop" value="stop" onclick="ShutMeDown()" id="BtnStop">
  </body>
</html>
```



Schwarz, klein und flach

Das BlackBerry Pearl 8100 ist ein Kommunikationstalent

Corinna Heyn



Es ist schwarz wie die Nacht und hat einiges zu bieten. Mit dem formschönen Smartphone können User kinderleicht im Internet surfen, telefonieren, SMS, MMS und E-Mails versenden sowie fotografieren. Der Clou an dem Gerät, das in jede Hemdentasche passt, ist der Trackball zur Navigation. Mit einer Hand kann mühelos zwischen den Menüpunkten hin- und hergesprungen oder mühelos im Internet gesurft und navigiert werden.

Das BlackBerry Pearl 8100 hat sogar einen USB-Anschluss. Da es sehr viele Funktionen bietet, gibt es einen Einrichtungsassistenten. Auf 15 Seiten werden alle Tasten, die Eingabemethoden, Schriftgrößen und die Einrichtung von E-Mail-Konten erläutert. Die ausführliche Einführung dauert zehn Minuten. Einfach das Handy einschalten und E-Mails schreiben, geht folglich

nicht. Dafür aber besteht die Möglichkeit, mit der BlackBerry Desktop Redirector-Option eine Verknüpfung zu Microsoft Outlook-E-Mail herzustellen.

Um eine E-Mail-Adresse einzurichten, muss der User zunächst ein „Neues Konto“ eröffnen, dann die Geschäftsbedingungen akzeptieren und eine ID plus Kennwort angeben. Mit dem Internet-Browser kann via Provider gesurft werden. Wenn ein Empfang da ist, geht das problemlos und schnell. Auch das Einloggen in Portale von Yahoo, um E-Mails zu schreiben, lesen oder zu beantworten, funktioniert



club|com|pu|ter.at; der größte unabhängige österreichische Computerclub mit über 1.000 Mitgliedern; die Geschichte des ClubComputer reicht bis ins Jahr 1986 zurück als die ersten Personal Computer leistungsfähig wurden; das Motto: „Wir begleiten unsere Mitglieder in die digitale Zukunft“

flyout.js

```
function Init()
{
    Station.selectedIndex =
        System.Gadget.Settings.readString("Station Index");
}

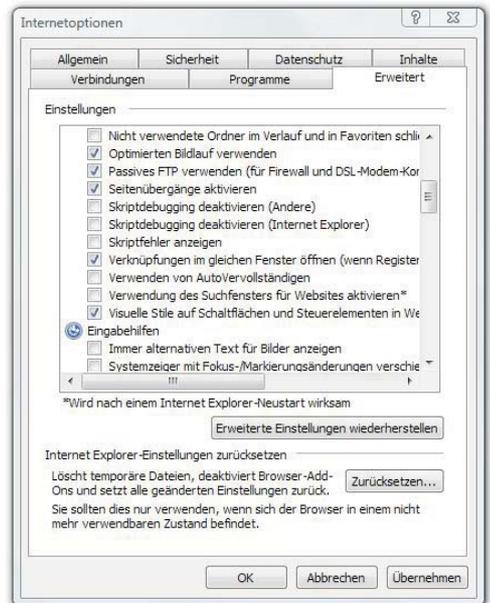
function WriteSettings ()
{
    System.Gadget.Settings.write (
        "Station Name", Station.options[Station.selectedIndex].text);
    System.Gadget.Settings.write (
        "Station URL", Station.options[Station.selectedIndex].value);
    System.Gadget.Settings.writeString (
        "Station Index", Station.selectedIndex);
    System.Gadget.Flyout.show=false;
}
```

Da man ein Flyout nur vom Haupt-Gadget (`atradio.htm`) aus- und einklappen kann, ich aber wollte, dass es sich automatisch einklapp, sobald ein neuer Sender ausgewählt wurde, verwende ich - wie übrigens viele andere Gadget-Developer auch - „Polling“, um zu sehen ob sich im Flyout etwas geändert hat. Das ist etwas „dirty“, lässt sich aber meines Wissens nach nicht besser lösen. Ich polle natürlich nur solange das Flyout offen ist, ob es einen neuen Sender in den `System.Gadget.Settings` gibt, wenn ja, starte ich diesen, und schließe das Flyout.

All das ist vom Haupt-Gadget aus notwendig und nur mit Polling möglich, es lässt sich nicht vom Flyout auf das Hauptgadget zugreifen!

Zum Testen muss man das Gadget immer wieder zur Sidebar hinzufügen bzw. schließen. Außerdem empfiehlt es sich, Skriptdebugging einzuschalten, damit man Javascript-Fehlermeldungen bekommt.

So sieht nun das fertige Gadget in der Sidebar aus, mit ausgeklapptem Flyout:



Ziemlich genial funktioniert das Packaging des neuen Gadgets: Einfach alle Files zippen, und die Endung von `.zip` in `.gadget` umbenennen!

Fertig!

Mein Gadget bekommt man unter <http://www.this.at/ATRadio07.zip>.

Man kann mein Gadget aber auch schon bei der Vista-Gallery <http://vista.gallery.microsoft.com/> herunterladen und bewerten.

Würde mich auch über Feedback freuen!