

Compatibiliteit beter geregeld in Windows 7

APPLICATIE PROBLEMEN IN DEVELOPMENT SNEL TE ONDERKENNEN

Paul van Wingerden

Tijdens het ontwikkelen van Windows 7 is er door Microsoft veel aan gedaan om de compatibiliteit te waarborgen. Het ontwikkelproces bevat automatische checks om te garanderen dat API's niet veranderen. Verder is er een keur aan tools, aandacht en informatie beschikbaar om applicatieproblemen zo vroeg mogelijk in de developmentcyclus te onderkennen. Dit gebeurt onder andere door middel van Telemetriegegevens die verzameld worden door bijvoorbeeld het Windows Quality Improvement program.

Programma's die getest worden zijn niet alleen applicaties maar ook bijvoorbeeld het .NET Framework, Java middleware enzovoorts. Verder wordt er intensief samengewerkt met OEMs om de met Windows 7 meegeïnstalleerde software goed te laten werken. Windows 7 is verder uitgerust om eindgebruikers in staat te stellen om compatibility-issues op te lossen. Een gecrashte applicatie kan door Windows 7 gedetecteerd worden en via het Action Center kan er gekeken worden of er een update beschikbaar is.

De overgang van Windows XP naar Windows Vista was een flinke stap; Vista was min of meer een nieuw Operating System. De verschillen tussen Vista en Windows 7 zijn veel kleiner; Windows 7 bouwt voort op de fundamenten van Windows Vista. Vista voegde zaken toe zoals de Aero Glass user interface, User Account Control, een nieuwe Installer maar ook zaken zoals de XML Paper specification. Veel van deze toevoegingen in Windows Vista zijn verbeterd of uitgebreid. Met de komst van Windows 7 zijn er weer een flink aantal nieuwe features bijgekomen zoals de Branchcache, Direct Access, Applocker, een nieuwe versie van Internet Explorer (8), PowerShell, group policies en nog veel meer.

Wellicht het grootste probleem waarmee developers geconfronteerd werden bij het uitbrengen van applicaties op Windows Vista was User Access Control (UAC). Veel applicaties waren nog niet helemaal klaar voor Vista. Er waren veel applicaties die niet in staat waren om met Least User Credentials (LUA) te werken. Feitelijk is dat niet een probleem van Windows Vista maar meer een consequentie van het niet opvolgen van best practices die al sinds Windows 2000 bestonden. Doordat de meeste gebruikers onder Windows XP als administrator werkten, kreeg Vista hiervan de schuld.

Hoe gaat dit met Windows 7 te werk? Het goede nieuws is dat application compatibility hetzelfde is als onder Windows Vista.

Dus indien een applicatie werkt onder Vista dan is de kans zo'n 99% dat die ook werkt onder Windows 7. Alleen als er sprake is van low level applicaties zoals drivers of virusscanners dan kan dit niet het geval zijn. Het 'slechte' nieuws is dat applicaties die op Windows Vista niet werken ook op Windows 7 niet zullen werken. Het is dus tijd om deze waar nodig aan te passen. Ik zou bijna zeggen: het is de hoogste tijd hiervoor. Gebruikers moeten zo veilig mogelijk kunnen werken en LUA is een van de technieken die hierbij helpen.

Belangrijkste veranderingen in Windows 7

Een van de meest in het oog springende veranderingen is het versienummer. De GetOSVersionEx call onder Windows 7 geeft '6.1' terug. Een van de meest voorkomende incompatibiliteiten tijdens het testen met Windows 7 was een hard gecodeerde versiecheck. De beste oplossing is om een check vanaf een bepaalde operating system versie te doen in plaats van een vaste check. Gebruik dus `>=` in plaats van `=`.

Het goede nieuws is dat we de applicatie kunnen laten geloven dat deze onder een andere Windows versie werkt. Dit kan met behulp van een techniek die een 'Shim' wordt genoemd en kan zonder het aanpassen van code. Er zijn vele typen Shims. Shims zijn er niet alleen om de applicatie om de tuin te leiden, maar bijvoorbeeld ook om 'deprecated API calls te loggen. Meer informatie is te vinden op <http://url.com>

Gebruik de operating system versie ook niet om te bepalen of een feature wel of niet aanwezig is. Een voorbeeld hiervan is bijvoorbeeld de aanwezigheid van Windows Media Center. Deze is niet aanwezig in alle versies van Windows 7. Het is mogelijk dat een gebruiker of een systeembeheerder deze feature niet heeft geïnstalleerd.

Geer dus niet bij voorbaat van uit dat het openen van een .msg file direct de mailapplicatie opstart. Er bestaat immers de kans dat de file extensie niet geregistreerd is of dat er geen mail applicatie

is geïnstalleerd. Als een applicatie een mail moet versturen, gebruik dan de APIs om te checken of de mail applicatie is geïnstalleerd. Welke applicaties dit zijn is aan de systeembeheerder of de gebruiker.

Internet Explorer 8 User Agent String

De User Agent String is een string waarmee Internet Explorer zich identificeert. Het verschaft informatie over het versienummer en een aantal andere attributen aan Web servers. Veel webapplicaties zijn afhankelijk van een bepaalde agent string versie en passen de html die naar de browser teruggestuurd wordt hierop aan. De User Agent string bevat nu de waarde 'Trident/4.0' om het verschil aan te geven tussen de Internet Explorer 7 user agent string en de Internet Explorer 8 User Agent string indien die in IE7 compatibility view mode draait. Verdere details zijn te vinden op: <http://url>. Een webpagina die expliciet checkt op een bepaalde versie van de User Agent string en die niet de IE8 User Agent string ondersteunt zal niet goed renderen. Veelal zal een gebruiker de pagina gewoon sluiten en een andere website proberen. Ga bij het testen van de User agent String dan ook uit van 'greater than' in plaats van 'equals'. Een volgende versie van Internet Explorer zal dan blijven werken en dit resulteert in minder onderhoud aan de html code. De Internet Explorer Compatibility Test Tool (IECTT) die met de Application toolkit meegeleverd wordt helpt om hierop te testen. Meer informatie over Internet Explorer 8 kunt U vinden op <http://www.microsoft.com/windows/internet-explorer/readiness/testing-tools.aspx>.

Het goede nieuws is dat we in Windows 7 de applicatie kunnen laten geloven dat hij onder een andere versie werkt.

User Access Control

Een van de grootste veranderingen tussen Windows XP en Windows Vista was de introductie van User Access Control. Ten tijde van Windows XP en voorgaande versies werkten de meeste gebruikers met administrative credentials. Deels omdat dit de standard instelling was, deels omdat het de gebruiker niet echt gemakkelijk gemaakt werd om met 'user credentials' te werken. Een voorbeeld hiervan betrof het veranderen van de Timezone, wat niets anders is dan het veranderen van de wijze waarop deze getoond wordt versus de datum/tijd die de systeemtijd daadwerkelijk veranderd. Het veranderen van de systeemtijd is een kritische instelling. Denk bijvoorbeeld aan auditing, immers wanneer de systeemtijd te manipuleren valt kan je bijvoorbeeld events 'veranderen'.

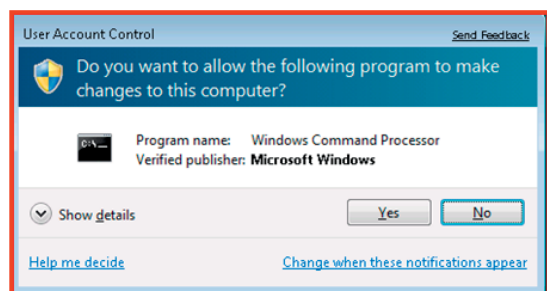
Het veranderen van de Timezone zit in dezelfde dialog verwerkt als het veranderen van de systeemdatum/tijd waardoor zelfs het kijken naar de systeemtijd betekende dat er administrator credentials nodig waren.

Indien men onder Windows 7 aanloopt als standard user (member van de users group) dan wordt er een logon sessie gecreëerd met alleen de meest basale permissies. De nieuwe sessie kan hierdoor geen instellingen wijzigen die het hele systeem betreffen. Als er nu

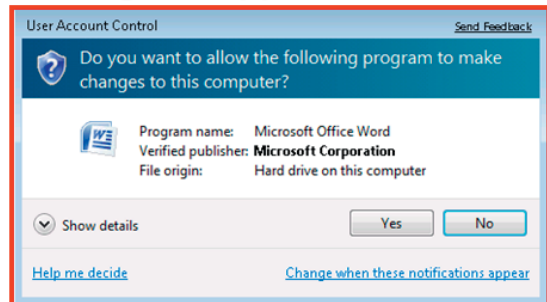
aangelogd wordt en de gebruiker is tevens member van de administrators group dan worden er twee tokens gecreëerd. Een token met de permissies die bij een administrator horen, en een tweede (restricted) token met dezelfde permissies als de hierboven beschreven user. Een applicatie wordt altijd opgestart met het restricted token.

Hierdoor kan de applicatie geen schade aanrichten zelfs niet als de gebruiker lid is van de administrators group. Als een applicatie meer permissies nodig heeft, bijvoorbeeld om een instelling te wijzigen, dan verschijnt er een UAC prompt. Indien de gebruiker hiervoor toestemming heeft (consent) dan wordt het proces gestart met de administrative permissies. De UAC prompt ziet er verschillend uit voor een user of een administrator. Indien een gebruiker alleen member is van de users group komt de volgende prompt naar voren:

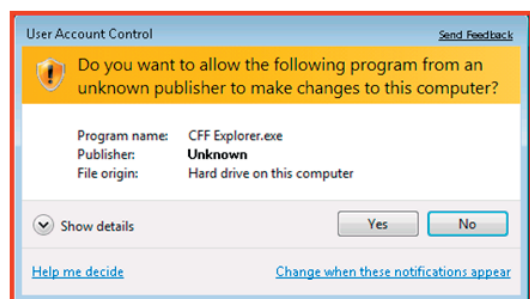
OS Application



Signed Application



Unsigned Application



Let op de verschillen tussen een signed-, een unsigned- en een OS applicatie. De gele band verschijnt bij een unsigned applicatie. Immers men weet niet waar die vandaan komt of door wie die is geschreven.

Indien een taak elevation vereist is dan dient dit in de user interface duidelijk zichtbaar te zijn. Hiervoor is er een icoon beschikbaar, zodat er een consistente look ontstaat. Dit icon moet de developer niet zelf tekenen maar bijvoorbeeld oproepen door middel

van het versturen van een message naar een button. Dit is het icoon dat dan op de button komt te staan:



Deze aanroep ziet er als volgt uit:

```
lResult = SendMessage( // returns LRESULT in lResult
    (HWND) hWndControl, // handle to destination control
    (UINT) BCM_SETSHIELD, // message ID
    (WPARAM) wParam, // = 0; not used, must be zero
    (LPARAM) lParam // = (LPARAM) (BOOL) fRequired;
);
```

In Windows 7 is een aantal kleine wijzigingen aangebracht in UAC. Een aantal meegeleverde onderdelen van Windows vereisen geen elevatie meer. Het nivo waarop UAC de prompt laat zien kan via het control panel worden aangepast.

Essentieel voor een goede analyse is om de unit tests te runnen die specifiek voor de applicatie in kwestie geschreven zijn.

Tools

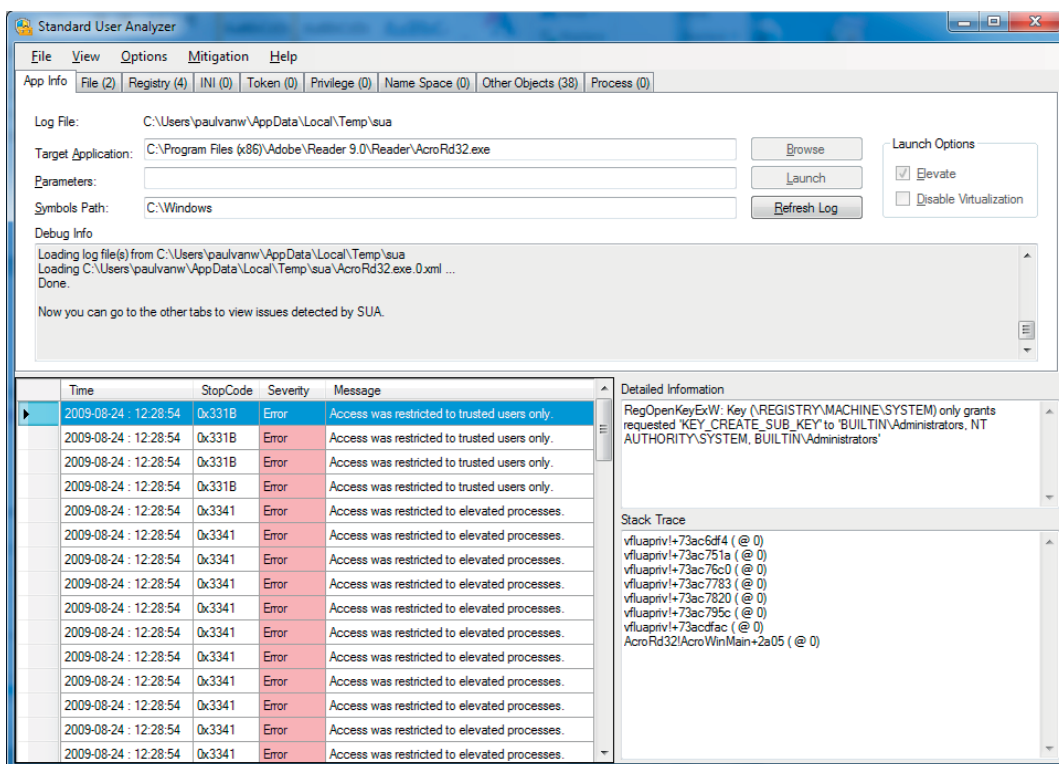
Application Verifier

De application Verifier tool helpt developers om subtiele programma fouten te vinden die gedurende gewone testen niet naar voren komen. De tool monitord de interactie van een applicatie met het operating systeem. Er wordt gekeken naar het gebruik van kernel objecten, de registry, filesystem en Win32 APIs (heap, handles, locks, etc.) De geprofileerde data wordt in een XML file weggeschreven en kan bijvoorbeeld met Excel geopend worden voor nadere analyse. De application verifier kan ook vanaf de command line gestart worden waardoor deze gebruikt kan worden in bijvoorbeeld een gautomatiseerde test nadat een build gedaan is. De XML file kan dan op fouten gescanned worden of vergetelen met de vorige build/

Je kunt een aantal tests configureren:

De Basic Verification Layer is de basislaag van de application verifier. Hier vinden checks plaats wat betreft de heap, structured exception handling, memory, locks, etc. De Limited User Predictir test hoe een applicatie zich gedraagt met user permissies (zie ook verder dit artikel). In de Miscleaneous verification layer zitten een aantal zeer interessante tests nl de dangerous API tests. Calls naar Loadlibrary en Freelibrary worden geanalyseerd en de Terminate thread call wordt aan een nadere analyse onderworpen. De Print Verification Layer tenslotte is ervoor gemaakt om fouten in het printing subsystem te vinden wanneer de applicatie deze aanroeft. Er zijn vele tientallen testen en condities waarop getest wordt. Meer informatie hierover kunt U vinden op: [http://msdn.microsoft.com/en-us/library/dd371695\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd371695(VS.85).aspx)

Omdat de tool een runtime testtool is verdient het aanbeveling een goede code coverage te hebben met de tests die in de applicatie uitgevoerd worden. Het meest voor de hand liggende scenario is om de unit testst te runnen die voor de applicatie in kwestie geschreven zijn. Dit is essentieel voor een goede analyse.



SCREENSHOT VAN DE STANDARD USER ANALYZER:

Standard user analyzer

De standard user analyzer is een tool die gebruikt kan worden door developers om te analyseren of een applicatie zich goed gedraagt als die met minimale permissies (user) wordt opgestart. De SUA tool is een runtime diagnose tool. Het kent twee modes, predictive en diagnostic. In predictive mode wordt de applicatie opgestart met administrative privileges. SUA monitored dan een groot aantal APIs die vooral resource access doen, zoals files en

Alle applicaties met een Windows Vista-certificering werken zonder problemen op Windows 7 en ook voor Windows XP-applicaties waren problemen snel op te lossen.

registrykeys. Tijdens het draaien van de applicatie wordt bijgehouden hoe iedere API aangeroepen wordt, het resultaat wordt geanalyseerd. Hierna wordt gelogd of de call geslaagd of niet geslaagd zou zijn als er met minimale permissies (user) gedraaid zou worden in plaats van administrator. Zodoende komen de meeste 'standard user' fouten boven water.

In diagnose mode wordt de applicatie opgestart met standaard user permissions. De eerste de beste fout lam de applicatie laten crashen. Dit is vooral nuttig nadat de applicatie eerst getest is en eventuele fouten verbeterd zijn door de SUA tool in predictive mode te draaien.

De standard user analyzer tool kan gedownload worden via: <http://www.microsoft.com/downloads/details.aspx?FamilyID=C4A25AB9-649D-4A1B-B4A7-C9D8B095DF18&displaylang=en>

De SUA tool is afhankelijk van de eerder genoemde application verifier.

Het Windows 7 Logo Programma

Het doel van het Windows 7 logo programma is om klanten te helpen bij het vaststellen of een product aan een aantal door Microsoft opgestelde tests voldoet. Het Windows 7 logo programma is in het bijzonder bedoeld voor Independent Software Vendors (ISVs). Deze mogen het logo voeren en op de productverpakking afdrukken. Voor een eindgebruiker geeft dit een garantie dat deze software technisch goed werkt op Windows 7. Er zijn ook bepaalde marketingvoordelen zoals application finder waar een applicatie in vermeld kan worden indien het logo behaald is.

Bij voorgaande edities van het Windows Logo programma bestond er een procedure met een extreme certificerende partij. De applicatie moest opgestuurd worden naar deze partij en enige tijd later kwam er feedback met wat wel en niet werkte. Hier ging meestal enige tijd overheen. Ook waren er kosten aan verbonden wat voor sommige klanten een belemmering bleek te zijn. De huidige editie van het logo programma voor Windows 7 is hierom

ook aangepast. Het bestaat uit een self test tool waarmee het logo aangevraagd kan worden. Hieraan zijn geen kosten verbonden. De self test tool bestaat uit een programma wat een aantal maken vanaf de commandline opgestart moet worden. Er zijn vijf momenten waarop dat moet gebeuren; de eerste maal voordat de applicatie is geïnstalleerd, daarna nadat de applicatie is geïnstalleerd maar niet opgestart, daarna kan de applicatie opgestart worden en kunnen er relevante tests mee gedaan worden (zie hiervoor ook de application verifier). Hierna moet de applicatie verwijderd worden en kan de laatste maal de test gedraaid worden. De test verzameld allerlei informatie over het systeem en slaat deze op. In de laatste stap wordt de informatie verzameld en in een XML bestand geplaatst. Dit bestand is in een leesbare vorm te krijgen door middel van een meegeleverde stylesheet. Het rapport dat hieruit komt geeft aanwijzingen wat er wel en niet aan het logo voldoet inclusief referenties naar de documentatie.

De laatste stap betreft het aanvragen van het logo. Ook indien het commercieel niet wenselijke geacht wordt is het draaien van de logo tool een best practice. Het verhoogt de kwaliteit van de software, het bespaart supportkosten voor de leverancier en vermindert de beheerkosten voor de eindgebruiker.

Meer informatie over het Windows 7 logo programma kunt U vinden op <http://msdn.microsoft.com/en-us/windows/dd203105.aspx>

Ervaringen en conclusie

Tijdens de Beta fase is er met een aantal klanten workshops gehouden om vast te stellen of en hoe goed de software werkt op Windows 7. De belangrijkste constatering was dat alle applicaties die een Windows Vista certificering hadden ondergaan zonder problemen werkten op Windows 7. Van de applicaties die getest zijn die onder Windows XP werkten (met of zonder certificering) waren de problemen makkelijk te identificeren en op te lossen.

Deze problemen bestonden voor 95% uit de volgende zaken:

- Applicatie data wegschrijven in ..\Program Files\<app> in plaats van in de ..\ProgramData\.. folder
- Schrijven van registrykeys in HKLM ipv HKCU
- Installeren in de root van C: in plaats van ..\Program Files\..

Allemaal problemen die Least User Access gerelateerd waren en die vrij eenvoudig op te lossen zijn. Een aantal van de geteste applicaties is inmiddels in het Logo programma opgenomen.

Op moment van het schrijven van dit artikel zijn er nog geen exacte data bekend wanneer er opnieuw workshops plaatsvinden. Houd daarom de blog van Mark Voermans in de gaten. Via <http://www.isvblogvoer.nl/> kunt U een aankondiging verwachten wanneer en waar er in de nabije toekomst een workshop georganiseerd wordt.



Paul van Wingerden, is Developer evangelist bij Microsoft. Zijn specialisatie: SQL Server, BI, web development, Windows Live. Paul is te bereiken op paulvanw@microsoft.com.