

Veel mensen hebben zich de afgelopen tijd afgevraagd hoe Java-software het nu precies op Windows Vista doet. Daar kunnen we kort over zijn: Java-software werkt uitstekend op Vista. Het volledige programmeerteam dat aan Java Platform Standard Edition werkt, was al bezig met het geschikt maken van Java-software voor Vista toen dit besturingssysteem nog naar een rundveesoort was vernoemd.

Java-software op Vista

Bugs en fixes

Tussen het regelmatig telefoneren met Microsoft, uitwisselingen met hun programmeurs in het geval van problemen of vragen, het over en weer rapporteren en herstellen van bugs en het volop testen en herschrijven van code tijdens de ontwikkeling van Vista door hebben we een uiterst solide versie van het Java-platform voor Vista weten te bouwen. Maar waarom zou men zich deze vraag überhaupt stellen? Bij oudere versies is er wel sprake van problemen met Vista, maar dat zou bij softwareontwikkelaars geen verbazing moeten wekken. Verwachten dat versies van Java-software voor eerdere versies van Windows automatisch op Vista zullen werken is net zoals verwachten dat je favoriete XBox-spel, Bloody Mess X, niet op de XBox360 werkt. Natuurlijk doet het dat niet. Het oorspronkelijke spel is voor een totaal ander systeem geschreven. Waarom zou je dan verwachten dat het spel kant-en-klaar geschikt is voor een nieuw platform? Het zou fantastisch zijn als je spel gewoon op het nieuwe systeem zou werken, maar daar was het simpelweg niet voor bedoeld. De platforms verschillen genoeg van elkaar om aanzienlijke wijzigingen van toepassingen te vereisen.

Naar alle waarschijnlijkheid, zeker gezien de populariteit van Bloody Mess X, zullen ontwikkelaars het spel naar de nieuwe console porten ("Nu nog meer bloed!"), zodat je in staat zult zijn om je favoriete spel ook op het nieuwe platform te gebruiken.

Hier hebben we met hetzelfde fenomeen te maken: Windows Vista is simpelweg geen XP++. Het platform bevat fundamentele verschillen die ervoor zorgen dat de oudere software het loodje legt. Betekent dit dat alle software niet meer werkt? Zo'n vaart zal het waarschijnlijk niet lopen. Maar hoe meer systeembronnen een toepassing gebruikt, des te waarschijnlijker is het dat deze tegen gewijzigde

systeemonderdelen aanloopt. Java-software is verre van een simpele Win32 GUI-toepassing; het is een runtime-platform met een diepgewortelde behoefte aan bronnen binnen het besturingssysteem, zoals de netwerk-stack, het beveiligingsmodel of het grafische systeem. Als een van deze subsystemen op ingrijpende wijze wordt veranderd, zullen we onze software overeenkomstig moeten aanpassen. In het geval van Vista omvatte dit een volledig proces van leren, testen, debuggen, het rapporteren van bugs bij Microsoft en deze stappen vervolgens weer opnieuw herhalen. Omdat Vista tijdens de ontwikkeling van Java Platform Standard Edition 6 (Java SE 6) een platform in beweging was, gaf elke nieuwe druppel Vista aanleiding tot een nieuwe ontwikkelingscyclus.

Het extra plezier dat we hebben beleefd aan het debuggen van een platform in beweging houdt echter niet in dat we alle problemen hebben opgelost. Tijdens de ontwikkeling van Java SE 6 vormde Vista een van onze hoogste prioriteiten, reden waarom de software al maandenlang uitstekend op Vista presteert. Een aantal van de meer in het oog springende bugs (zoals onze deactivatie van de Aero Glass-effecten) hebben we hersteld. De fixes zijn sinds de lente van 2006 beschikbaar in snapshot-versies van Java SE 6. Daarnaast hebben we tijdens het herstellen van problemen in Java SE 6 (ons primaire platform voor Vista-ondersteuning) de belangrijkste fixes met terugwerkende kracht naar oudere versies geport. Hiermee zijn we in staat om updates te bieden aan klanten die hun oudere Java-versies ook graag op Vista zien draaien.

Plezier

Als dit artikel een white paper zou zijn, zou deze waarschijnlijk een titel als "Java-software werkt prima op Vista" hebben. Vervolgens zou het docu-

Chet Haase
architect van de
client-group van Sun

In sommige gevallen vertoonden Java-GUI's plotseling allerlei zwarte blokjes of gaven ze **content op de verkeerde plaats weer**

ment het telefoonnummer en e-mailadres vermelden van een PR-vertegenwoordiger die u precies hetzelfde zal zeggen, alleen in veel meer woorden. De meeste ontwikkelaars zijn echter meer geïnteresseerd in een aantal technische details waarmee we moesten stoeien om onze Java-software op Vista aan de praat te kunnen krijgen. Ik zal hier geen volledige waslijst van problemen en fixes opvoeren, maar daarentegen een paar van de belangrijkste punten aanstippen.

Aero Glass

Het "Aero Glass"-bureaublad wordt geactiveerd door de Desktop Window Manager (DWM) van Vista. Dit systeem draagt zorg voor de grafische weergave van het bureaublad (vensters, pictogrammen, kortom, alles wat er op het scherm te zien valt) en draait op Direct3D (een runtime-library die ondersteuning biedt voor driedimensionale grafische bewerkingen met hardwareversnelling). Alle "vensters" die op het bureaublad worden afgebeeld, zijn in feite samengestelde texture-maps. Dit systeem maakt verschillende beeldscherm-effecten mogelijk, zoals doorzichtige titelbalken, voorbeeldweergaven van toepassingen op de taakbalk en nieuwe Alt-Tab- en Cmd-Tab-functionaliteit waarmee gebruikers tussen toepassingen kunnen schakelen.

Probleem 1: Deactivatie van DWM

* *Symptoom*: Tijdens het uitvoeren van oudere versies van Java-software kon het voorkomen dat de DWM op non-actief werd gesteld, waarmee effectief het Aero Glass-effect werd gedeactiveerd. Hierdoor bemerkten gebruikers een wijziging van de bureaubladweergave: de vensterranden werden ondoorzichtig, een aantal hippe bureaubladeffecten ging verloren en vensters van toepassingen begonnen meer op Windows XP-vensters te lijken. Bij oudere versies was dit waarschijnlijk het meest in het oog springende probleem qua compatibiliteit van Java-software en Vista, aangezien gebruikers werden geconfronteerd met een plotselinge en duidelijke wijziging van de bureaubladweergave.

* *Probleem*: Java 2D maakt gebruik van DirectX voor de weergave en versnelling van afbeeldin-

gen en buffers buiten het scherm. Om verschillende redenen (zoals een statuscontrole om te zien of DirectX nog actief was en grafische bewerkingen synchroniseerde) waren we gedwongen om het scherm te "vergrendelen" om rechtstreekse toegang tot de pixels op het beeldscherm te krijgen. Het vergrendelen van het beeldscherm is echter een uiterst effectieve manier om de DWM van Vista op non-actief te stellen als gevolg van de nieuwe samengestelde bureaubladaanpak van Vista. De toepassingen van de DWM hadden op deze manier niet langer rechtstreeks toegang tot het beeldscherm. Wanneer een toepassing (zoals via deze oudere DirectDraw-vergrendelingsopdracht) om toegang vraagt, zal Vista dit verzoek inwilligen, maar pas nadat het de DWM heeft gedeactiveerd, en daarmee dus ook Aero Glass.

* *Oplossing*: We hebben minder voor de hand liggende manieren gevonden om onze zin te krijgen zonder het beeldscherm te vergrendelen. Zo wordt de DWM niet uitgeschakeld en kan de Java-software vlekkeloos op Vista draaien.

Probleem 2: Grafische objecten

* *Symptoom*: Zodra we de bug hadden opgelost waardoor de DWM werd uitgeschakeld, kwam een ander probleem aan het licht. In sommige gevallen vertoonden Java-GUI's plotseling allerlei zwarte blokjes of gaven ze content op de verkeerde plaats weer.

* *Probleem*: Desktop Java gebruikt zowel DirectX als GDI om content binnen vensters van Java-toepassingen weer te geven. Dit werkte uitstekend in het geval van eerdere Windows-versies, maar zorgde voor een puinhoop in Vista.

* *Oplossing*: De snelste oplossing was om in Vista simpelweg niet langer gebruik te maken van DirectX. In feite waren we gedwongen om een keuze te maken tussen GDI en DirectX. We konden niet buiten GDI, omdat we toegang tot systeembronnen voor vensters nodig hebben. DirectX viel daarom af. Er is een veel completere oplossing in de pijplijn voor Java Platform Standard Edition 7 (Java SE 7) die een veel vollediger implementatie bovenop Direct3D biedt en een meer betrokken en vaardig buffersysteem dat

ons in staat zal stellen om DirectX op een veel Vista-vriendelijker manier te gebruiken.

Native look & feel van Swing

Vista heeft een volledig nieuwe look & feel. Naast de reeds besproken Aero Glass-effecten zijn de knoppen, vensters en alle andere reguliere bureaublad-widjets een nieuw uiterlijk voorzien. Vista heeft niet alleen de bronnen voor de look & feel van XP getweaked, het nieuwe uiterlijk zelfs op basis van een volledig nieuw mechanisme geïmplementeerd. De oude manier waarop Swing gebruikmaakte van de bronnen voor de look & feel-bronnen van XP was daarom aan herziening toe.

Probleem 1: Nieuwe native look & feel

* *Symptoom*: Een Swing-toepassing die gebruikmaakt van de native look & feel van oudere versies van Java-software krijgt ook een native look in Vista. Het is alleen wel de native look van Windows 2000.

* *Probleem*: In Windows XP ontleed Swing zijn native look & feel aan de systeembronnen van XP. Kort gezegd gebruikten we dezelfde systeembronnen als XP om onze componenten van een native look te voorzien. Vista maakt daarentegen gebruik van een volledig nieuw systeem met nieuwe bronlocaties en formaten. In Vista ontbreekt het ons daarom niet alleen aan een native Vista-look; we hebben zelfs geen XP-look.

* *Oplossing*: In Java SE 6 is de native look & feel van Swing voor zowel Windows als GTK op zodanige wijze herschreven dat voor de weergave van Swing-widjets de native rendering-engine wordt gebruikt. In plaats van onze oude methode om zelf de weergave te verzorgen, roepen we nu het native rendering-systeem aan om de widjets voor ons op het scherm te tekenen (via de UXTheme-API). Zodra ons werk was voltooid, zag Swing er native Vista uit in Vista (en native XP in XP).

Probleem 2: Native look & feel van de animatie

* *Symptoom*: Sommige Vista-onderdelen gebruiken animatie-effecten om een overgang van de ene naar de andere status aan te duiden. Als de muis bijvoorbeeld over een knop wordt bewogen, zal een crossfade worden uitgevoerd om de overgang van de normale status op de mouse-over-status aan te geven. Swing-componenten (net als de native bureaubladonderdelen van eerdere Windows-versies) schakelden echter altijd tussen statussen zonder animatie-effect. Waar binnen een native Vista-toepassing een animatie zou worden weergegeven, zou in het geval van een Swing-toepassing een simpele (niet-geanimeerde) statuswijziging te zien zijn.

* *Probleem*: Bij eerdere versies van Windows was er geen sprake van een animatiesysteem. Onderdelen wijzigden simpelweg van status en werden op het scherm afgebeeld op een manier die met de nieuwe status overeenkwam. Om ruimte te bieden aan het nieuwe animatiegedrag, moesten we nieuwe logica implementeren waarmee het uiterlijk van toepassingen na een bepaalde gebeurtenis kon worden gewijzigd.

* *Oplossing*: Recente builds van Java SE 6 bieden basisondersteuning voor Swing-animatie. In sommige gevallen komt het uiterlijk nog niet volledig met dat van Vista overeen, maar over het algemeen zien de zaken er goed uit. Er zijn verdere fixes gepland voor updates van Java SE 6 en Java SE 7.

Installatie

Een van de belangrijkste wijzigingen binnen Vista was van invloed op onze installatietechnologie. Het beveiligingssysteem van Vista is ten opzichte van eerdere Windows-versies op belangrijke punten gewijzigd om het malware moeilijker te maken om de controle over de systeembronnen over te nemen. Gebruikers worden nu bijvoorbeeld op de hoogte gesteld wanneer een toepassing toegang zoekt tot systeembronnen als het register. Ze zullen een dialoogvenster te zien krijgen dat hen om goedkeuring voor de bewerking in kwestie vraagt en naar het beheerderwachtwoord vraagt als ze niet als beheerder zijn aangemeld. Internet Explorer 7 (IE7) doet hier nog een schepje bovenop door het volledige proces de toegang tot de ruwe systeembronnen te ontzeggen. Dit houdt in dat toepassingen die vanuit de browser toegang tot het systeem krijgen, alleen in staat zijn om bewerkingen uit te voeren binnen de beperkte sandbox (beschermde ruimte) die door de browser ter beschikking wordt gesteld.

Deze wijzigingen hadden verstrekkende gevolgen voor een aantal van onze installatiecomponenten, zoals de Installer, Java Update, Java Plug-in en Java Web Start. Omdat er bij veel van deze problemen sprake was van een overlap, zal ik me beperken tot het bespreken van de algemene probleemgebieden waarvoor we een oplossing moesten vinden.

Probleem 1: Beheerdersrechten vereist

Er zijn verschillende situaties waarin we beheerdersrechten nodig hebben om een benodigde installatietaak uit te voeren. Voorbeelden zijn het opslaan van informatie in registersleutels, het schrijven naar locaties op de harde schijf en het uitvoeren van processen.

* *Symptomen*: De belangrijkste gevolgen van deze wijzigingen waren dat sommige onderdelen van

het installatie- of verwijderingsproces niet naar behoren functioneerden en dat gebruikers steeds om toestemming moest worden gevraagd voor het uitvoeren van bepaalde taken (zoals Java Update).

* *Probleem:* Vista probeert om de toegang tot systeembronnen zoveel mogelijk af te schermen. Sommige processen (zoals installatieprocessen) hebben “extra rechten” nodig waarvoor de toestemming van de gebruiker is vereist. In sommige gevallen voerden we een installer als een afzonderlijk proces uit, hetgeen mislukte doordat er geen mogelijkheid was om de gebruiker om de vereiste toestemming te vragen. Op vergelijkbare wijze had Java Update extra rechten nodig om te kunnen controleren of er updates beschikbaar waren. Dit proces vond telkens plaats wanneer een gebruiker zich aanmeldde, hetgeen op den duur eerlijk gezegd op de zenuwen begon te werken.

* *Oplossing:* We hebben manieren bedacht om de installatieproblemen te omzeilen. Omdat er voor het probleem met Java Update geen eenvoudig antwoord beschikbaar was, hebben we simpelweg de frequentie beperkt waarmee dit proces wordt uitgevoerd.

Probleem 2: De sandbox van IE7

IE7 probeert alle mogelijke beveiligingsrisico's af te dekken door de toegang tot het bestandssysteem vanuit het browserproces te beperken. Als gevolg hiervan wordt de Java Plug-in, die in de vorm van een ActiveX-control in IE7 wordt uitgevoerd, volledig binnen de sandbox van IE7 vastgehouden. Hierdoor worden de systeemlocaties waaruit we moeten lezen of waarnaar we moeten schrijven, volledig onbereikbaar.

* *Symptoom:* De Java Plug-in was niet in staat om JAR-bestanden te cachen en andere gegevens met Java Web Start en Java Control Panel te delen.

* *Probleem:* Java Web Start en het Java Control Panel worden als standalone processen uitgevoerd, en hebben daarom geen problemen met het lezen uit, of schrijven naar de benodigde schijflocaties (zoals een reguliere installatiemap). De Java Plug-in kan daarentegen geen toegang tot deze mappen krijgen en zou daarom naar een nieuwe

en gevirtualiseerde schijflocatie moeten schrijven, die echter weer niet zichtbaar zou zijn voor de andere toepassingen, waardoor het niet mogelijk zou zijn om gegevens tussen deze toepassingen te delen.

* *Oplossing:* Recentere Vista-builds bieden een gepersisteerde map die voor alle processen (al dan niet beperkt) bereikbaar is. Deze map bevindt zich binnen het sandbox-gebied waarnaar IE7 kan schrijven. Hierdoor blijft het bestandssysteem beveiligd, en zolang andere, niet-IE7-processen (zoals Java Web Start en het Java Control Panel) deze locatie kunnen bereiken, zullen we in staat zijn om alle benodigde gegevens te delen. Door deze nieuwe map als gedeelde installatielocatie te gebruiken, zullen Java-toepassingen in staat zijn om alle informatie als vanouds te delen.

Probleem 2A: Vista en IE7-sandboxes

Er is sprake van een onoplosbaar dilemma dat de moeite van het opmerken waard is: in Vista geeft IE7 ons geen toegang tot het bestandssysteem buiten de IE7-sandbox voor applets. Dit betekent dat zelfs applets die van een handtekening zijn voorzien geen rechten hebben om naar willekeurige locaties op de harde schijf te schrijven. Dit gedrag wijkt af van het vertrouwde applets-gedrag in eerdere versies van Windows, maar vanwege de beperkingen van IE7 bestaat er voor dit probleem geen enkele workaround. Toepassingen zullen zich naar deze nieuwe situatie moeten schikken en naar locaties binnen Vista moeten schrijven die wel gebruikbaar zijn.

Er is sprake van een verwant probleem dat eveneens de moeite van het aanstippen waard is. Vista heeft bepaald dat er naar verschillende mappen simpelweg niet mag worden geschreven. Zo is het niet langer mogelijk (of in ieder geval niet langer eenvoudig) om bestanden naar de root-map (C:\) te schrijven of naar andere mappen op systeemniveau. Niet-beheerders zijn normaliter alleen in staat om naar hun thuismap te schrijven. Deze wijziging is niet alleen van invloed op applets, zoals in het geval van de eerder besproken beperkingen op het gebied van de

Lees verder op pagina 57

in Vista geeft IE7 ons geen toegang tot het bestandssysteem buiten de IE7-sandbox voor applets

Vervolg van pagina 8

IE7 sandbox, maar eveneens op standalone toepassingen. Dit is natuurlijk geen probleem dat specifiek op Java-software van toepassing is, maar wel een probleem waarmee gebruikers van onze oplossingen mee moeten leren leven, net zoals de gebruikers van andere native toepassingen. Bestanden moeten en zullen worden opgeslagen in Vista-vriendelijke locaties.

Het plan

Veel mensen die dit artikel lezen zullen na dit overzicht van bugs en fixes met een aantal vragen blijven zitten, zoals: "Wanneer kan ik een versie van Java-software krijgen die goed op Vista werkt?" Dat is een uitstekende vraag, en ik ben blij dat u die heeft gesteld. Hier volgt een overzicht van onze plannen (pin ons niet op de onderstaande data vast: de data zijn met zorg op onze planningen en verwachtingen gebaseerd, maar u begrijpt waarschijnlijk wel dat data voor software-releases zich net zo snel kunnen bewegen als u bij ze in de buurt komt als vliegen bij uw lunch):

** Java SE 6 is de beste oplossing voor Vista:* Ons primaire softwarevehikel voor Vista is Java SE 6; deze versie heeft tijdens de ontwikkeling van Vista onze hoogste prioriteit gekregen en is voorzien van de meeste fixes voor bekende problemen. We hebben zojuist de laatste hand aan deze versie gelegd en zullen deze later dit jaar uitbrengen. In de tussentijd raden wij u aan om de download-website voor Java SE 6 (<http://jdk6.dev.java.net>) te bezoeken en de laatste snapshot op te halen en te testen. Alle belangrijke fixes voor problemen met Vista werden enige maanden geleden in deze release opgenomen, waardoor deze een uiterst geschikt uitgangspunt vormt voor de implementatie van Java-software op Vista.

** Java Platform Standard Edition 5.0 (Java SE 5.0) zal ook werken:* Veel van de Vista-fixes zijn reeds met terugwerkende kracht naar Java SE 5.0 geport, of zullen dat binnenkort worden. Verwacht echter niet dat alles op exact dezelfde wijze zal werken. Onze primaire prioriteit was namelijk dat Java SE 6 de Vista-release zou worden. Het meeste Vista-werk dat naar onze mening voor Java SE 5.0 is vereist zou in update 11 aanwezig zijn. Deze versie hopen wij ergens vroeg in 2007 uit te brengen. Mogelijk volgen er nadere Vista-fixes in toekomstige updates van Java SE 5.0.

** Java Platform Standard Edition 1.4.2 (Java SE 1.4.2) zal gewoon werken:* We zijn van plan om de nodige fixes met terugwerkende kracht naar Java SE 1.4.2 te porten nadat we dit voor Java SE 5.0 hebben gedaan. Mijn waarschuwing met betrekking tot Java SE 5.0 geldt ook voor Java SE

1.4.2. Sterker nog, we zijn van plan om alleen de strikt noodzakelijke problemen te herstellen zodat Java SE 1.4.2 op Vista kan draaien, en niet al teveel tijd te besteden aan het verbeteren van de basisfunctionaliteit. We beschouwen Java SE 1.4.2 als functioneel, bruikbaar en uiterst geschikt voor situaties waarin gebruikers volledig van deze specifieke versie afhankelijk zijn. Desondanks moedigen we ontwikkelaars en klanten die Vista gebruiken uit om binnenkort te migreren naar een Java SE-versie met volledige functionaliteit.

Conclusie

Ik hoop dat dit artikel ons verhaal de stand van zaken bij Sun een beetje heeft verduidelijkt. Ik ga in meer detail in op dit onderwerp op mijn weblog: <http://weblogs.java.net/blog/chet/>. Mijn dank gaat uit naar het gehele Java SE-team bij Sun, zonder wiens intensieve programmeer- en testwerk dit alles niet mogelijk zou zijn geweest. Werkt u met Windows Vista, schroom dan niet om Java SE 6 (een pre-release snapshot te downloaden van <http://jdk6.dev.java.net> <<http://jdk6.dev.java.net/>> of de volledige versie, wanneer beschikbaar, te downloaden van <http://java.com> <<http://java.com/>> of <http://java.sun.com> <<http://java.sun.com/>>) en in Vista uit te proberen.