

De focus van Java heeft vooral gelegen op het ontwikkelen van grote Enterprise applicaties met lompe applicatieservers en een grote verzameling frameworks. In december 2008 is de eerste versie van JavaFX vrijgegeven om terrein terug te winnen op de desktop en in de browser. Ondertussen zijn we na een hobbelige weg bijna drie jaar verder en staat de release van JavaFX 2.0 gepland voor eind dit jaar. Wat mogen we ervan verwachten? Wat verandert er ten opzichte van wat we al kennen van JavaFX?

De nieuwe features van JavaFX 2.0

Definitieve versie komt voor einde van het jaar

Inmiddels heeft Oracle de Beta van JavaFX gereleased. Dit stelt jou als lezer in de gelegenheid om ook zelf met de nieuwe features van JavaFX 2.0 te gaan spelen. De Beta en de documentatie zijn te vinden op⁹.

JavaFX? Wat is dat?

Mocht JavaFX in het geheel aan je voorbij zijn gegaan, geen nood, laten we beginnen bij het begin. In december 2008 werd de eerste versie van JavaFX in de wereld losgelaten. Het idee achter JavaFX is het opnieuw leven inblazen van Java op de desktop. Met een geheel nieuwe, declaratieve scriptingtaal (JavaFX Script) werd het ineens ontzettend makkelijk om mooie gebruikersinterfaces te maken. De eenvoud waarmee animaties en transities gemaakt werden had iets weg van Flash of Flex.

Daarnaast introduceerde de taal diverse krachtige constructies om veranderingen in data automatisch gevolgen te laten hebben in de interface. Het bind-keyword stelde je niet alleen in staat om de waarde van een property te koppelen aan de waarde van een variabele. Je kon er complete connecties mee maken van bijvoorbeeld de X-coördinaat van de muis met de opacity van een Button op je scherm.

De originele belofte van JavaFX omvatte niet alleen de desktop op de computer. Ook integratie in de browser, een mobiele runtime en zelfs de televisie. Helaas werd al snel duidelijk dat veel van die belofte niet zou worden gehaald. Versie 1.0 ondersteunde alleen de desktop en de browser. Vanaf versie 1.1 konden JavaFX applicaties ook op een mobieltje met Java ME draaien. De TV development kit werd gereleased met versie 1.3, maar er zijn (nog) geen officieel ondersteunde televisies.

Ondanks deze obstakels is JavaFX toch een interessant platform voor het ontwikkelen van grafisch hoogstaande applicaties en het visualiseren van (realtime) data. Het is ook in een aantal projecten gebruikt waar men anders wellicht C# of Flex had gekozen, vooral omdat een JavaFX interface rechtstreeks vanuit Java code gevoed kon worden. Die integratie tussen 'vanilla' Java en JavaFX Script zorgde ervoor dat je in principe een bestaande applicatie kon pakken en er een JavaFX frontend aan kon hangen.

JavaFX 2.0

Vorig jaar creëerde de overname van Sun door Oracle een onzekere tijd voor JavaFX-ontwikkelaars. Het duurde tot de zomer voordat Oracle officieel aankondigde het platform door te ontwikkelen en JavaFX 2.0 op onze radar zette. Er werd echter ook een aantal drastische wijzigingen aangekondigd. Oracle ziet de potentie van een platform op Java dat zich kan meten met Flash/Flex en Silverlight, maar ziet ook dat de adoptie van de JavaFX Scripttaal minimaal is. Daardoor wordt de kans op slagen van het platform aanzienlijk kleiner.



Jan Hendrik Kuperus
is Java Technology Specialist
bij Presenter.

```
Stage {
  title: "Say Hello!"
  scene: Scene {
    content: [
      Text {
        y: 10
        x: 20
        content: "Hello world!"
      }
    ]
  }
}
```

Hello world in JavaFX Script.

Vandaar dat in JavaFX 2.0 deze extra drempel wordt verwijderd. Alle APIs zijn ondertussen omschreven naar 'normale' Java. Dit stelt in één klap het platform beschikbaar aan alle ontwikkelaars op de JVM, inclusief Scala, Groovy en vele andere talen.

Er kleeft alleen één groot nadeel aan deze keuze: mensen die met het huidige JavaFX Script applicaties hebben gemaakt, moeten opnieuw tijd steken in het omschrijven van hun werk. Niet alles is verloren, aangezien de mensen achter het platform ook alles moeten herschrijven en hier een tooltje voor hebben ontwikkeld. Bij de Early Access release, die we eerder dit jaar konden bewonderen in een NLJUG university sessie, zat een conversie-tool. Deze schrijft het meeste automatisch om naar de nieuwe APIs. Bij de huidige Beta blijkt deze tool echter te ontbreken. Wellicht dat men van plan is deze als een losse tool aan te bieden, in plaats van te bundelen met de JavaFX 2.0 SDK.

Een alternatief om alsnog de oude JavaFX Script syntax te kunnen gebruiken wordt geboden door het Open Source project Visage. Hierin wordt door onder andere Stephen Chin een poging gedaan om de 'oude' syntax in leven te houden en zelfs compatible te maken met JavaFX 2.0 code.

Binding

In JavaFX Script heb je de beschikking over het bind-keyword, wat het koppelen van de text-property van bijvoorbeeld een Text-component aan een variabele extreem eenvoudig maakt.

```
var myTextInput : String;

def textBox = Text {
    content: bind myTextInput
};

myTextInput = "Hello World";
```

JavaFX Script Binding.

Een groot voordeel van de keuze om voor normale Java te gaan, is dat de kracht van het bind-keyword is omschreven naar een Java library. Dat betekent dat we er een (hopelijk) goed binding framework bij hebben gekregen dat ooit zijn weg in Java SE zal vinden. In de huidige Beta vinden we hiervan echter nog maar beperkte voorbeelden. In het package javafx.beans.binding vinden we de basis binding classes en een factory class Bindings, die de meest voorkomende binding expressies kan opbouwen. Vreemd genoeg worden deze classes in geen enkele sample gebruikt.

In plaats daarvan wordt er veel gebruik gemaakt van de interface InvalidationListener. Dit ziet eruit als een meer low-level API voor het maken van bindings. Een InvalidationListener krijgt namelijk een update zodra de onderliggende waarde verandert.

Bij het gebruik van een binding wordt door ditzelfde mechanisme de binding als dirty aangemerkt, waardoor een volgende raadpleging van de binding de waarde opnieuw wordt berekend. Dit lazy binding mechanisme zou volgens de documentatie de performance van de applicatie ten goede komen. Als dat zo is, dan is het op zijn zachtst apart te noemen dat de samples hier geen gebruik van maken.

```
StringProperty myTextInput = new StringProperty();

Text textBox = new Text();
textBox.contentProperty().bind(myTextInput);

myTextInput.set("Hello World");
```

JavaFX 2.0 Binding.

To Build or to Swing?

Met het opgeven van de declaratieve taal om JavaFX user interfaces te bouwen rijst de vraag hoe we dan nu onze UI in elkaar moeten gaan zetten. In de aanloop naar het Early Access programma is hier flink over gediscussieerd, zowel intern als ook extern zoals bijvoorbeeld op de Birds of a Feather sessie over JavaFX op Devoxx vorig jaar. Men twijfelde sterk over het gebruik van het Builder-pattern, of om het 'gewoon' op de Swing manier te doen.

Het Builder-pattern heeft als voordeel dat het redelijk dicht bij de look en feel van het originele scripttaaltje komt. Zoals je in onderstaand voorbeeld kunt zien, lijkt er weinig te zijn veranderd in vergelijking met het eerste voorbeeld. Een tweede voordeel is dat het gebruik van builders het voor andere talen op de JVM eenvoudig maakt om dezelfde declaratieve syntax aan te houden. Sinds de aankondiging van JavaFX 2.0 circuleren al diverse voorbeelden van Groovy over het internet die amper van JavaFX Script zijn te onderscheiden. Hoewel zelfs met de release van de Beta nog niets zeker is, lijken we met aan zekerheid grenzende waarschijnlijkheid te kunnen zeggen dat Builders onderdeel uit blijven maken van de nieuwe JavaFX API. Het complete package javafx.builders is gewijd aan deze manier van het opbouwen van je interface.

```
Stage s = new Stage();
s.setTitle("Say Hello!");
s.setScene(new SceneBuilder()
    .root(new TextBuilder()
        .y(10)
        .x(20)
        .content("Hello world!")
        .build()) // TextBuilder
    .build()); // SceneBuilder
```

Builder Pattern JavaFX.

Hoewel het er op Devoxx vorig jaar op leek dat men een keuze ging maken tussen Builders en Swingachtige APIs, bleek op de University Sessie bij Oracle eerder dit jaar dat beide alternatieven zich in de Early Access builds bevonden. De redenatie die hierbij werd gegeven is dat op deze manier de drempel ook voor huidige Swing-ontwikkelaars

**Builders
blijven
onderdeel
uitmaken
van de
nieuwe
JavaFX API.**



**NIEUWE
WORKSHOP
MET
DR. BAS VAN
GILS**

3 november 2011 • Hotel Lapershoek Hilversum

Succes met Enterprise Architectuur

U MAG DEZE WORKSHOP NIET MISSEN!

- Gebruik van TOGAF en ArchiMate
- Grip krijgen op de organisatie
- Makkelijk aanpasbaar aan lokale behoeften
- Effectief omgaan met veranderende omstandigheden



Enterprise architectuur heeft een belangrijke opmars gemaakt in organisaties. Als discipline wordt het architectuurdenken op verschillende niveaus in organisaties ingezet. Architecten komen in aanraking met verschillende disciplines, van strategie en business modeling, tot procesmanagement, informatiemanagement, IT, en infrastructuur. Een succesvolle architectuurfunctie geeft grip op de organisatie, en geeft antwoord op de vraag "hoe moeten we ons organiseren?" De noodzaak om effectief om te gaan met steeds veranderende omstandigheden, zowel binnen als buiten de organisatie, staat hierbij centraal.

De workshop

Architectuur is een praktisch en multidisciplinair vakgebied. Deze workshop bevat veel korte en praktische oefeningen. Bij uw online aanmelding kunt u aangeven bij welke organisatie u werkt, wat uw organisatie doet aan architectuur en hoeveel ervaring u al met architectuur heeft. In kleine groepjes worden de oefeningen voorbereid en centraal besproken. Hierdoor krijgt u voldoende houvast om het geleerde meteen toe te passen in de eigen praktijk.

Bestemd voor ú

Deze workshop is geschikt voor een breed publiek. Bent u business consultant, (aspirant) architect, manager, verander- en informatiedeskundige of IT-er en bent u betrokken bij de professionalisering van de architectuurfunctie? Dan mag u deze workshop niet missen!

**KIJK SNEL OP WWW.ARRAYSEMINARS.NL
VOOR MEER INFORMATIE!**

wordt verlaagd. Het komt ook direct ten goede aan de doelstelling van Oracle om JavaFX met Swing te laten samenwerken. Volgens de roadmap (te vinden op javafx.com/roadmap) is ook te zien dat deze Swing Interoperability af is en dat Swing en JavaFX componenten door elkaar kunnen worden gebruikt.

```
Scene theScene = new Scene(new Text(10, 20,
    "Hello World"));
```

```
Stage stage = new Stage();
stage.setTitle("Say Hello!");
stage.setScene(theScene);
```

Swing-achtige JavaFX.

De voorbeelden die bij de Beta zitten laten in ieder geval ook hier weer een interne voorkeur van Oracle zien. Geen enkele sample maakt gebruik van de Builders. Alles wordt opgebouwd volgens het Swing-idioom. Zeer waarschijnlijk komen er nog wel voorbeelden met Builders bij volgende builds van de Beta, maar het geeft ons een kleine blik in de gedachten van de mensen die aan het platform werken. Blijkbaar worden de Builders vooral beschouwd als hulpmiddelen om de declaratieve manier van instantieren te vereenvoudigen voor alternatieve talen.

Hardware Acceleratie

Van de speerpunten voor JavaFX 2.0 moet de nieuwe Prism pipeline zorgen voor ongeëvenaarde grafische performance. Op de keynote op JavaOne van 2010 werd een eerste demonstratie van deze nieuwe pipeline gegeven. Deze begon met wat grappige effecten en Flash-achtige animaties, maar eindigde met een behoorlijk indrukwekkende scene. In de laatste scene werden maar liefst 160 verschillende filmpjes tegelijk afgespeeld in een interactieve movie-wall. Deze 160 '2D' filmobjecten werden ook nog met een 'eenvoudige' 3D-translatie in de ruimte weergegeven.

Al met al gaf dit een erg leuk visueel effect en demonstreerde dit de potentiële kracht van de nieuwe renderengine. De belofte is dan ook dat scenes met 15.000 nodes en meer geen enkel probleem meer mogen zijn in JavaFX 2.0. In eerdere versies liep je al snel tegen een degradatie van het aantal frames per seconde aan, vooral als je erg veel bindings ging gebruiken.

De Prism engine gaat afhankelijk van je platform gebruik maken van DirectX of OpenGL. Op Windows systemen zal de Prism engine draaien op DirectX 9 (WinXP & Vista) of DirectX 11 (Win7) en op Mac en Unix zal gebruik worden gemaakt van OpenGL. Mocht hardware acceleratie om wat voor reden dan ook niet aanwezig zijn op het doelplatform, dan valt Prism automatisch terug op Java2D, maar daarmee wordt dus wel flink op de performance ingeleverd.

JavaFX en het web

JavaFX moest vanaf het begin eigenlijk al de alom bekende (of juist wel beruchte) Applets vervangen op het web. Java zou weer het platform worden om leuke dingetjes mee op het web te maken. Daarom waren de JavaFX runtime libraries online beschikbaar en werden applicaties eenvoudig opgeleverd met behulp van WebStart. Helaas bleek ook dit niet altijd even goed te werken. Iedere JavaFX WebStart-applicatie werd namelijk afhankelijk van de bereikbaarheid van de website www.javafx.com en dat kon nooit lang goed gaan. Toen begin 2010 deze website er een dag uit lag werd meteen duidelijk hoe fragiel deze oplossing is.

De nieuwe manier van WebStart-deployment zal dan als het goed is ook niet langer deze afhankelijkheid hebben. Of dit ook gaat lukken moeten we even afwachten. Zolang JavaFX niet standaard in de Java runtime aanwezig is, zal er waarschijnlijk toch vaak nog iets gedownload moeten worden. Opname in Java 7 gaat in ieder geval nooit gebeuren en Java 8 is nog maar afwachten.

Met de huidige opkomst van HTML5 en CSS3 krijgt JavaFX het ook nog best moeilijk om belangrijk op het web te worden. Gelukkig hebben ze daar ook iets op verzonnen: JavaFX heeft een meegeleverde, op WebKit gebaseerde browser. Hierin kan HTML5-content worden getoond in een JavaFX-applicatie. Zo bestaat er beide kanten op een manier om met JavaFX dingen met het web te doen. JavaFX-applicaties kunnen in een webpagina worden opgenomen en JavaFX-applicaties kunnen HTML-content bevatten. Nu is het wachten op de eerste applicaties die op deze manier een oneindige loop met zichzelf maken.

The Verdict

JavaFX heeft vanaf het begin een wat rommelige reputatie. Langs halve releases en instabiele APIs zijn we bij in april 2010 met versie 1.3 op een werkbare versie uitgekomen. Met de komst van 2.0 moeten we deze ellende achter ons laten en met verse moed dit nieuwe platform voor rich applications oppakken. De grote vraag is of dit ook op grote schaal gaat gebeuren. Gaan we bijvoorbeeld nu dan echt Java games zien? Blijft het bij wat Flash-achtige spelletjes? Het is lastig om dit nu te voorspellen. De Beta ziet er aardig stabiel uit en alle beloofde functionaliteit zit er ook in.

Wat dat betreft heeft de overname van Sun door Oracle tot nu toe in ieder geval duidelijkheid qua deadlines opgeleverd. Zowel de Early Access als de Beta waren beide beschikbaar op de eerder beloofde tijdstippen. Het is dan ook zeer waarschijnlijk dat we voor het einde van het jaar echt met JavaFX 2.0 aan de slag kunnen. «

**JavaFX
krijgt het
nog moeilijk
om belang-
rijk op het
web te
worden.**

Zelf aan de slag

Mocht je nu alvast een kijkje willen nemen, download dan de Beta op www.javafx.com en start dan in ieder geval de sample Ensemble.jar een keer. Heb je nog 'oude' JavaFX Script code liggen, kijk dan ook zeker eens naar het Open Source project Visage.