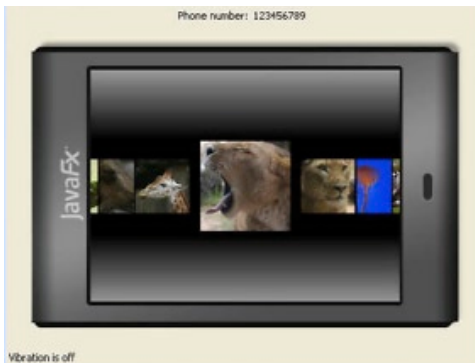


APEX 'across all screens' met JavaFX

APEX wordt met elke nieuwe release beter en verrijkt met nieuwe features. Werd APEX tot een paar jaar geleden nog als niet meer dan een Excel of Access vervanger afgedaan, met het groeiende palet aan nieuwe functionaliteiten hebben we nu de mogelijkheid om APEX voor een breed scala aan toepassingen in te zetten. Door APEX te combineren met een technologie als JavaFX kan een verscheidenheid aan toepassingen worden gebracht op een groot aantal platformen. Van smartphone tot desktop, van tablet tot tv, met behulp van JavaFX maken we makkelijk mooie toepassingen die gebruiksvriendelijk en grafisch aantrekkelijk zijn.

De afgelopen jaren, en met name het afgelopen jaar, hebben laten zien dat er nog genoeg ruimte is voor revolutie in de digitale wereld. Begin 2010 lanceerde Apple de iPad. Een nieuw marktsegment was geboren. Ook al wisten we initieel niet wat we er mee moesten, één ding is nu duidelijk, iedereen wil tegenwoordig een tablet hebben.

Naast de iPad en andere tablets is ook de smartphone serieus begonnen aan een opmars. Eind vorig jaar bestond dertig procent van de mobiele telefoons in Nederland uit smartphones. Bij een smartphone gaat het al lang niet meer om het kunnen bellen met het apparaat. Het is voor velen van ons een integraal onderdeel van ons leven geworden; internet, bedrijfsmail en de talloze social networks waar we ons in begeven.



De Mobile emulator van JavaFX, ontwikkelen voor je mobiel op je pc.

Deze ontwikkelingen geven wel aan dat er een enorme behoefte is om op elk moment van de dag informatie tot ons te kunnen nemen. Hoewel tablets in eerste instantie als consumentenproduct werden gezien, begint het concept nu ook in het bedrijfsleven aan terrein te winnen. En waarom ook niet? In 2010 begonnen we al steeds meer diensten en informatie in Cloud-omgevingen te hangen en deze trend lijkt in 2011 in een stroomversnelling te raken. Zware desktops en laptops zijn dus niet altijd meer nodig. Hardware bepaalt niet langer de norm; een goede verbinding met internet is belangrijk.

Op al onze apparaten zit tegenwoordig de mogelijkheid verbinding te maken met het internet en op al deze apparaten willen we ook gebruik maken van 'internet behoevende' applicaties.

- Er zijn vanuit het perspectief van web-based datacentrische applicaties heel wat toepassingen te bedenken voor deze mobiele platformen. Een paar voorbeelden;
- Stel je wilt als manager vlak voordat je een business meeting in gaat nog snel een laatste update van de voortgang van je unit hebben. Snel kijk je op je smartphone waar de grafieken realtime data weergeven en je in één oogopslag de meest actuele situatie schetsen. BI-light.
 - Of je bent een bodemonderzoek aan het doen. Op je tablet zie je een dwarsdoorsnede van de grond en zie je de coördinaten waar je een monster moet nemen. Dit alles dankzij GPS en Oracle Spatial.
 - Wellicht kan een huisarts in de toekomst wel bij een huisbezoek het hele dossier bekijken en bijwerken (en een recept meteen doorsturen naar de apotheek) op een tablet.

Ondertussen zijn er talloze slimme mobile devices met schermformaten van 3 tot 11 inch. Naast deze slimme apparaten zijn er natuurlijk ook nog de grote LCD tv's. Voor dit platform zijn dus ook de nodige toepassingen te bedenken.

Voor een bedrijf als Oracle, specialist in het ontsluiten en verwerken van data, zijn dit gouden tijden; al deze nieuwe platformen bieden mooie kansen tot de ontwikkeling van nieuwe toepassingen voor manipuleren en presenteren van data.

Binnen het team dat ontwikkelt aan Oracle APEX moeten ze

dit ook hebben gedacht want voor de komende release staan er wat zaken op het programma om APEX toegankelijker te maken op mobile devices. Dit beperkt zich met name tot aangepaste themes en templates. Op zich heel fijn, maar zoals alle APEX-gebruikers wel weten zijn de standaard layouts van APEX niet echt aantrekkelijk en is aanpassen niet eenvoudig. Een andere en misschien betere manier om APEX goed te kunnen gebruiken op je tablet of smartphone is door te integreren met een technologie die op het gebied van grafische user interfaces flexibeler is dan APEX. Eén mogelijke kandidaat hierbij is integratie met Oracle's eigen JavaFX.

Waarom JavaFX?

Het grote voordeel van JavaFX is de flexibiliteit waarmee mooie user interfaces kunnen worden gemaakt. Met JavaFX zijn we in staat om user interfaces te maken voor mobiles, desktop en browser. In de toekomst komen hier nog andere platformen bij zoals blu-ray en tv. Omdat JavaFX een hoop standaardfunctionaliteit biedt kunnen relatief eenvoudig complexe grafische interfaces worden gebouwd. Dit biedt ontwikkelaars veel mogelijkheden om snel gebruiksvriendelijke applicaties te bouwen. De slogan 'write once, run everywhere' geeft duidelijk het doel aan dat JavaFX poogt te bereiken met hergebruik van bestaande code en oplossingen die zo veel mogelijk toepassingen bedienen. Kort de voordelen van JavaFX op een rij:

- Mooie grafische interfaces voor (web)applicaties
- Ondersteuning voor Java, waardoor een heel arsenaal aan frameworks ter beschikking staat aan de ontwikkelaar
- Gericht op hergebruik van code

Integratie van JavaFX in APEX

Integratie binnen APEX is slechts een deel van het verhaal. Het tonen van een Java-applet op een webpagina is niet al te moeilijk. Zeker met de nieuwe plug-in feature van APEX 4 is het aanroepen van JavaFX voor APEX-ontwikkelaars een koud kunstje. De kracht van de oplossing is dat het voor beide kanten, APEX en JavaFX, voordelen oplevert. Voor APEX geldt dat JavaFX een scala aan interactieve mogelijkheden biedt die je met een declaratieve webapplicatie niet gauw zult kunnen nabouwen. Omdat JavaFX zowel binnen de browser als los op verschillende soorten apparaten kan draaien, heeft de gebruiker voortaan overal en altijd beschikking over data uit de Oracle database met een gebruikerservaring die ver boven een webapplicatie uitsteekt.

Voor JavaFX is APEX een laag tussen de front-end en de database: APEX voorziet in alle 'moeilijke' functionaliteit als queries uitvoeren, business rules valideren etc. De communicatie tussen de front-end en APEX-laag gebeurt door middel van open standaarden als http en XML, waardoor het een open structuur blijft. Dit voorkomt allerlei beveiligingsproblemen omtrent



JavaFX biedt een ruimte hoeveelheid aan controls, die zelf naar smaak te stylen zijn.

het toegang geven tot de database vanuit JavaFX en de complexiteit van een alternatieve laag tussen database en JavaFX. Kortom, beide technieken kunnen zonder elkaar. Maar dankzij de aanvullende mogelijkheden die ze elkaar bieden geven ze gezamenlijk enorme mogelijkheden.

Wat er nodig is om JavaFX te integreren met APEX hangt af van het type JavaFX-applicatie. Moet je deze data in de database stoppen of is alleen lezen voldoende? Moet de applicatie stand-alone draaien of moet deze geïntegreerd zijn in de APEX-pagina?

Afhankelijk van de antwoorden op deze vragen dient de APEX-applicatie ondersteuning te bieden. Voor het lezen uit de database kan gebruik worden gemaakt van communicatie via XML: APEX genereert op verzoek van JavaFX een XML-bestand. Dit kan door bijvoorbeeld een AJAX-callback (APEX4) of application process te gebruiken. De volgende code geeft hier een voorbeeld van, waarbij een XML-bestand met links wordt gegenereerd:

```
declare
v_blob blob;

begin
selectxmlelement ("document "
```



Voorbeeld van de integratie van een JavaFX applicatie: de gebruiker kan door afbeeldingen lopen.

```

        , xmlelement("links",
xmlagg(xmlelement("link",link)
))
).getblobval(873)
intov_blob
fromtab_links;

owa_util.mime_header('application/xml', false);

-- Set the size so the browser knows how much to download
htp.p ('Content-length: ' || dbms_lob.getlength( v_blob));
htp.p ('Expires: ' ||TO_CHAR(SYSDATE + 50, 'Day, DD Month, YYYY
HH24:MI:SS'));

-- The filename will be used by the browser if the users does a
"Save as"
htp.p ('Content-Disposition: attachment; filename="links.xml"');

-- Close the headers
owa_util.http_header_close;

-- Download the BLOB
wpg_docload.download_file ( v_blob);

end;

```

Door deze code op te nemen in een process kan JavaFX deze aanroepen via een URL. Bijvoorbeeld:

```

http://<pad_naar_apex>/f?p=<applicatie_nr>:<pagina
nr>:<sessie>:APPLICATION_PROCESS=<naam proces>

```

Door in de url items te refereren kunnen er parameters worden meegegeven, bijvoorbeeld:

```

http://<pad_naar_apex>/f?p=<applicatie_nr>:<pagina
nr>:<sessie>:APPLICATION_PROCESS=<naam proces>::PI_ID:1

```

Dit zorgt ervoor dat het item PI_ID de waarde 1 krijgt. Op basis van deze waarde kan vervolgens een XML-bestand worden gegenereerd. Om JavaFX in de pagina te integreren, moeten we HTML-code genereren waardoor de browser de JavaFX-applicatie gaat laden. Binnen APEX4 hebben we hiervoor de mogelijkheid een plugin te maken, waar je in APEX 3 met een HTML-region zou moeten werken. Een plugin binnen APEX bestaat uit een renderfunctie die de HTML genereert en een aantal attributen of eigenschappen. De volgende code geeft een voorbeeld van de renderfunctie:

```

function render
( p_region          in apex_plugin.t_region
, p_plugin          in apex_plugin.t_plugin
, p_is_printer_friendly in boolean
) return apex_plugin.t_region_render_result is

v_retval apex_plugin.t_region_render_result;

begin

htp.p('<script src="http://dl.javafx.com/1.2/dtfx.js"></script>');

htp.p('<script>
javafx
( { archive: "' || p_region.attribute_01 /* JAR-FILE */ || "'
, code: "' || p_region.attribute_02 /* CODE */ || "'
, name: "' || p_region.attribute_03 /* NAME */ || "'
},

```

```

{ server: "' || owa_util.get_cgi_env('http_host') || "'
, app_id: "&APP_ID."
, sessionid: "&SESSION."
, pageid: "&APP_PAGE_ID."
}

);
</script>);

return v_retval;

end;

```

In deze code worden drie attributen gerefereerd, waarin achtereenvolgens de locatie van het JAR-bestand, de code om op te starten en de naam van de applicatie staan. Zo weet de browser hoe hij JavaFX moet opstarten. Deze gegevens zijn te vinden in het JNLP-bestand. Dit bestand dient naast de JavaFX-applicatie te worden aangeboden aan de browser, het wordt meegeleverd met de JavaFX-applicatie. Het is dus handig deze twee bestanden op de webserver te zetten, onder bijvoorbeeld de bekende /i/ directory.

Om informatie in de database te kunnen opslaan vanuit JavaFX kan gebruik worden gemaakt van de standaard APEX-functionaliteit. Bijvoorbeeld Forms. Door een pagina te maken in APEX met een formulier en de daarbij behorende processen kan JavaFX de velden 'invullen' en opsturen naar APEX. Hierbij doet JavaFX hetzelfde als de browser zou doen bij het opsturen van het formulier. Daardoor maakt JavaFX gebruik van dezelfde componenten als de gebruiker. Dit levert winst op, omdat zodoende op een veilige manier via standaard APEX-componenten wordt gewerkt, waarmee JavaFX communiceert.

A winning Team

In dit artikel hebben we laten zien hoe eenvoudig het is om JavaFX te integreren in APEX. Door de kracht van APEX, de Oracle database en grafische mogelijkheden van JavaFX te combineren kan de toepasbaarheid van beide technologieën fors worden vergroot. Deze combinatie van technieken stelt ons in staat om voor alle populaire platformen (tablets, smartphones, laptops, etc) robuuste, grafisch aantrekkelijke en gebruiksvriendelijke applicaties te ontwikkelen. Dit biedt zowel APEX als JavaFX een mooie toekomst!



Bart Meijer is Systeemontwikkelaar bij de Caesar Groep en werkt als technisch consultant met APEX. Hij geeft training APEX 4.0 advanced. Hij is te bereiken op b.meijer@caesar.nl



Andrik Eker is Senior Oracle Consultant en teamleider van APEX specialisten bij de Caesar Groep. Hij is te bereiken op a.eker@caesar.nl