

Kennismaken met Oracle ADF Faces

Heeft Java eindelijk zijn Oracle Forms?

In een serie van drie artikelen geeft Lucas Jellema een rondleiding door de wereld van Java Server Faces. Dit tweede artikel is gewijd aan de Oracle JSF-implementatie: ADF Faces. We kijken onder meer naar de ingebouwde AJAX-faciliteiten van ADF Faces. Ook bespreken we de data-binding faciliteiten van ADF Faces. We gebruiken Oracle JDeveloper 10.1.3.2 als IDE. Hoewel enige Java-achtergrond nuttig is, zijn de artikelen toegankelijk voor alle Oracle-ontwikkelaars.

Het eerste deel gaf een introductie op de basiselementen binnen de JSF-standaard. Het volgende deel gaat in op andere JSF-componentenlibrary's zoals Apache MyFaces en ICE Faces, het combineren van verschillende library's en de ontwikkeling van eigen JSF-componenten.

Wat is ADF Faces?

ADF Faces is gewoon een implementatie van de JSF-standaard. Een J(2)EE webapplicatie kan (deels) op basis van Java Server Faces worden ontwikkeld en daarbij kan geheel of gedeeltelijk gebruik worden gemaakt van ADF Faces. Een ADF Faces applicatie kan ontwikkeld worden in een willekeurige Java IDE en gedeployed worden op iedere J(2)EE compliant application server.

ADF Faces is ook een bijzonder rijke, goeddoordachte bibliotheek van aantrekkelijke user interface componenten die de meeste aspecten van data-intensieve, administratieve applicaties goed ondersteunen. ADF Faces is de opvolger van Oracle UIX (User Interface XML) dat al sinds 2001 in JDeveloper beschikbaar is en door Oracle is gebruikt voor de ontwikkeling van de self-service componenten van de E-Business Suite. In zekere zin heeft ADF Faces een historie en een volwassenheid die de leeftijd van de JSF-standaard zelf ver overstijgt!

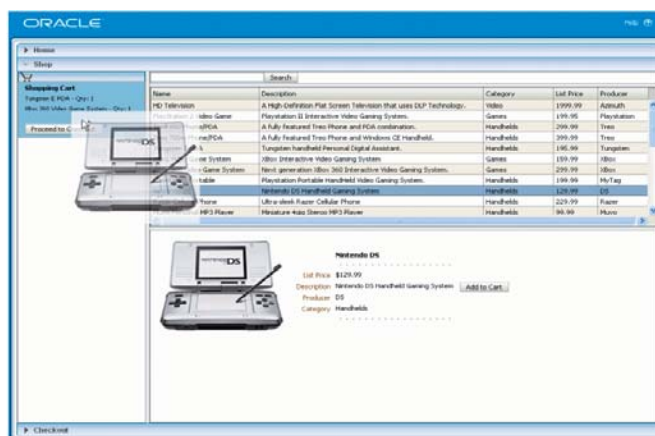
Eind 2005 – tijdens de JavaPolis conferentie in Antwerpen – kondigde Oracle aan ADF Faces beschikbaar te stellen aan de open-sourcegemeenschap. Nagenoeg alle broncode van ADF Faces werd overgedragen aan de Apache Software Foundation,

om uiteindelijk als MyFaces Trinidad gepubliceerd te worden. Dit betekent dat (vrijwel) alle functionaliteit van ADF Faces in het open-sourcedomein beschikbaar is en dus zonder verdere licentiekosten kan worden ingezet. Overigens is ook het gebruik van JDeveloper is gratis. Voor toepassing in een productiesysteem van ADF Business Components of het ADF Model is wel een runtime licentie – of gebruik van de Oracle Application Server – vereist.

ADF Faces wordt inmiddels op grote schaal gebruikt, overal ter wereld. Het is een belangrijke pijler onder het succes van Java Server Faces. De grootste gebruiker van ADF Faces is overigens Oracle zelf: de Oracle E-Business Suite en ook het aanstormende Fusion Applications zal op grote schaal van ADF en dus ook van ADF Faces gebruikmaken. Daarmee zijn de toekomstvastheid en zeker ook de verdere ontwikkeling van geavanceerde features – denk bijvoorbeeld Web 2.0 en RIA-gegarandeerd. Overigens is de recente EBS Release 12 nog op UNIX gebaseerd.

Rich Client Components in ADF Faces 11g

De volgende generatie van ADF Faces dient zich al aan: de rich(er) client components. Hoewel de huidige ADF Faces

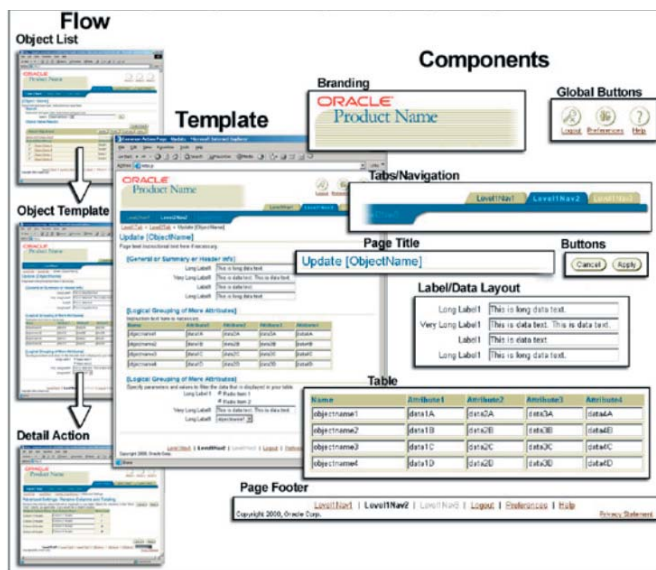


Abbeelding 1. Client Side Drag & Drop in de ADF Faces 11g Rich Client Components.

componenten al Ajax ondersteunen, staat de overtreffende trap qua rijkheid en dynamiek al trappend in de coulissen. De ADF Rich Client Components bieden functionaliteit als de accordion, client-side drag & drop, dynamisch configureerbare tabellen (verander kolommen van breedte en positie) en rechtermuis-knop pop-ups. Typische RIA- en Ajax-producten als het Nederlandse Backbase voelen hiermee de hete adem in hun nek.

BLAF

UIX en nu dus ook ADF Faces zijn zelf weer gebaseerd op Oracle BLAF (Browser Look & Feel Guidelines). In de tweede helft van de jaren negentig van de vorige eeuw gaf Larry Ellison opdracht om een Usability laboratorium op te zetten, waarin onderzoek moest worden gedaan naar de optimale user interface voor onder andere webapplicaties. Een flink aantal onderzoekers, onder meer aangetrokken van diverse universiteiten, onder leiding van Dan Rosenberg, heeft vele duizenden uren onderzoek gedaan, bijvoorbeeld met complexe video-opstellingen, om te komen tot een set aanbevelingen voor met name de structuur, opbouw, kernelementen en look & feel van HTML-applicaties. Deze set heet Oracle BLAF (zie voor meer details [4] in het kader 'Referenties' aan het eind van dit artikel).



Afbeelding 2. Enkele voorbeelden van BLAF suggesties in de traditionele Look & Feel.

Toen BLAF vaste vorm begon aan te nemen volgde een oekaze aan alle productontwikkelgroepen binnen Oracle: jullie webapplicaties moeten de voorschriften van BLAF volgen. Dat bleek makkelijker gezegd dan gedaan – de usability experts waren lang niet allemaal zelf ook HTML-ontwikkelaar! Om de ontwikkelaars binnen Oracle in staat te stellen op een productieve wijze te ontwikkelen binnen de aanwijzingen van BLAF werd een framework ontwikkeld dat uiteindelijk UIX ging heten. UIX

bodde de componenten die de look & feel in HTML konden realiseren en bovendien over de voorzieningen beschikte die de ontwikkelaar nodig had om functionele pagina's te ontwikkelen, zoals data binding en user-actie (button, hyperlink) verwerking.

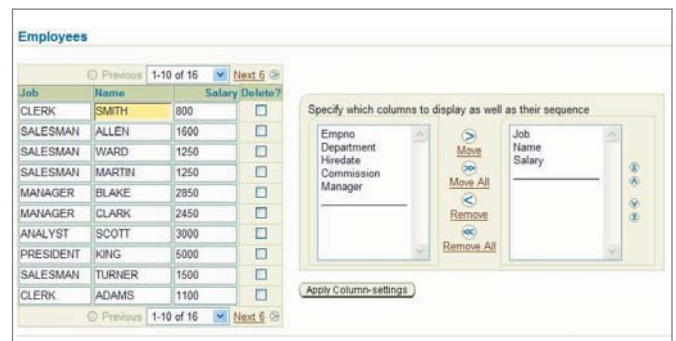
ADF Faces biedt nu diezelfde ondersteuning voor Oracle BLAF. Tegelijkertijd kunnen de ADF Faces componenten als zelfstandige eenheden worden gebruikt binnen een pagina die naar eigen inzicht wordt vormgegeven. Bovendien kan met het zogenaamde 'skinning' de look & feel volledig op maat worden gemaakt. Daarover later meer.

ADF Faces Componenten

De ADF Faces bibliotheek bevat meer dan honderd componenten, van bijna triviaal simpel tot zeer complex. Naast ADF Faces varianten met wat meer functionaliteit van de elementen uit de standaard JSF-set, zijn er handige componenten voor File Upload en Download, selecteren van een datum, ondersteunen van een List of Values venster en het netjes uitgelijnd tonen van een lijst van velden en prompts. Voor de organisatie van de componenten in de pagina is een reeks container-elementen beschikbaar, waaronder tabbed panels.

De Table van ADF Faces is het mooiere en slimmere broertje van de HtmlDataTable van de JSF-standaard. De Table ondersteunt Sorteerbare Kolommen, Pagineren van Rijen (toon rij X tot en met Y van Z), Insert/Update/Delete en de zogenaamde Detail Disclosure waarbij een rij kan worden 'opengeklapt' om extra details te tonen, behorend bij die rij. Meerdere tabellen kunnen genest zijn – om bijvoorbeeld complexe Master-Detail(-Detail...) relaties te presenteren. Onder de geavanceerde componenten vinden we onder andere de Train – voor het ontwikkelen van wizards –, de Shuttle – voor Multi-select operaties – en de Tree (en TreTable) – voor weergave en manipulatie van hiërarchische gegevensverzamelingen.

Een ADF Faces component aan een pagina toevoegen is ook weer een Drag & Drop operatie: selecteer de component in het ADF Faces componentenpalet en sleep deze naar de juiste plaats in de pagina of in het Structure Window. Voor veel componenten opent een wizard die een eenvoudige configuratie



Afbeelding 3. De ADF Faces Table en Shuttle componenten.

van de soms tientallen component-eigenschappen ondersteunt. Het ontwikkelen van een ADF Faces pagina gebeurt verder in de WYSIWYG editor en de ontwikkelaar wordt geholpen door het Structure Window – dat de boomstructuur van de pagina met alle elementen toont en waarin exacte positiebepaling mogelijk is – en de Property Editor waarin eigenschappen kunnen worden gedefinieerd op typisch 4GL-achtige wijze met dropdown lijstjes, hint texten en pop-up vensters. Het is opmerkelijk overigens hoe weinig ‘code’ een ontwikkelaar nodig heeft om tamelijk geavanceerde functionaliteit te realiseren. Neem bijvoorbeeld een webpagina met een editable overzicht van boeken, afkomstig uit een Collection van Book objecten, met sorteerbare kolommen, paginering, row-banding en een detail-disclosure.

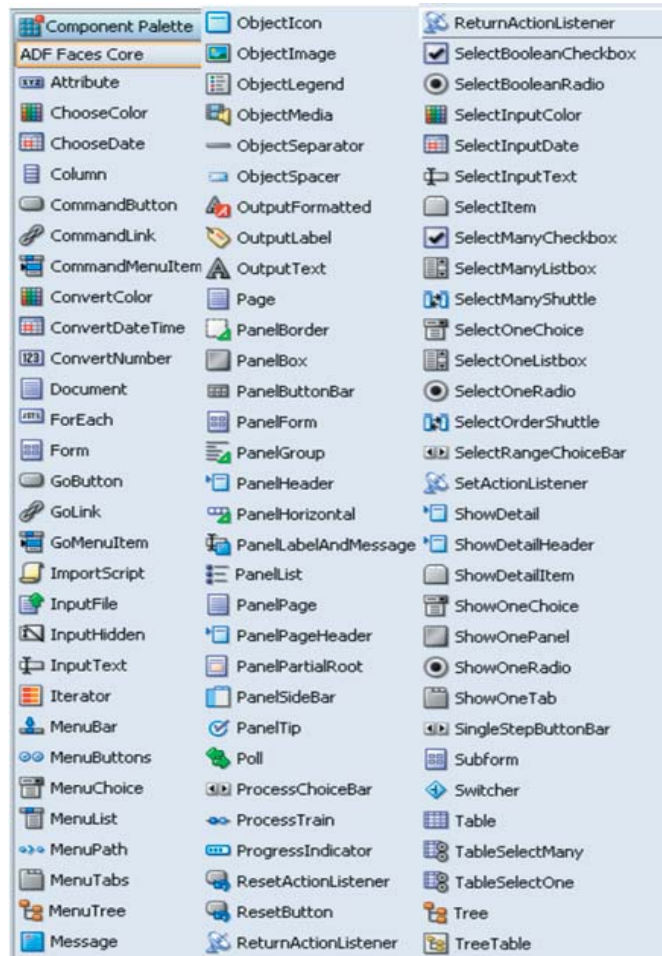
| Select Details | Title | Author | Publication Year | Publisher |
|--------------------------|-------------------------------------|------------------------------------|------------------|--------------------------|
| Show | JAVASERVER FACES IN ACTION | Kito D. Mann | 2004 | Manning Publications Co. |
| Show | HIBERNATE IN ACTION | Christian Bauer and Gavin King | 2004 | Manning Publications Co. |
| Hide | AJAX IN ACTION | David Crane | 2005 | Manning Publications Co. |
| Homepage ISBN 1932394613 | | | | |
| Show | ART OF JAVA WEB DEVELOPMENT | Neal Ford | 2003 | Manning Publications Co. |
| Show | PRO JSF AND AJAX: BUILDING RICH | Jonas Jacobi and John Fallows | 2005 | APress |
| Show | FOUNDATIONS OF AJAX | Ryan Asleson, Nathaniel T. Schutta | 2005 | APress |
| Show | BEGINNING POJOs. FROM NOVICE | Brian Sam-Bodden | 2004 | APress |
| Show | COST-BASED ORACLE FUNDAMENTALS | Jonathan Lewis | 2005 | APress |
| Hide | EXPERT ORACLE DATABASE ARCHITECTURE | Tom Kyte | 2005 | APress |
| Homepage ISBN 1590595300 | | | | |
| Show | EXPERT ORACLE DATABASE 10G | Sam R. Alapati | 2005 | APress |

Afbeelding 4. De Boeken Overzicht pagina met de tabel in actie.

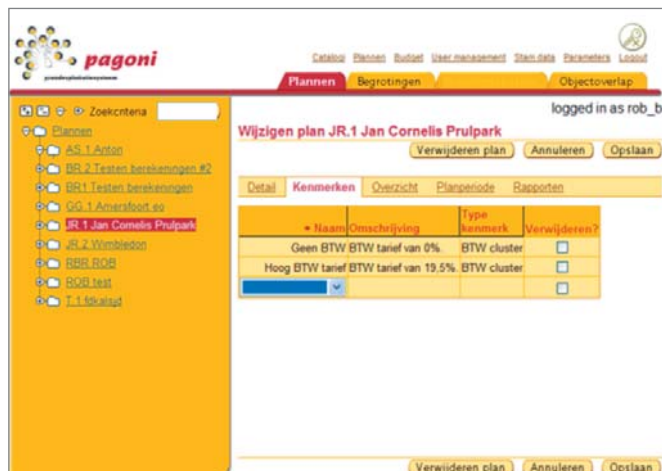
Voor de implementatie van deze component is niet meer code nodig dan dit JSF-fragment:

```
<af:table value="#{libraryManager.books}" var="book" rows="10"
    banding="row" bandingInterval="1">
    <f:facet name="selection">
        <af:tableSelectOne/>
    </f:facet>
    <af:column sortProperty="title" sortable="true" headerText="Title"
        formatType="text">
        <af:inputText value="#{book.title}"/>
    </af:column>
    <af:column sortProperty="author" sortable="true" headerText="Author"
        formatType="text">
        <af:inputText value="#{book.author}"/>
    </af:column>
    <af:column sortProperty="publicationYear" sortable="false"
        headerText="Publication Year" formatType="text">
        <af:inputText value="#{book.publicationYear}" columns="5" />
    </af:column>
    <af:column sortProperty="publisher" sortable="true"
        headerText="Publisher" formatType="text">
        <af:goLink text="#{book.publisher}"/>
    </af:column>
    <f:facet name="detailStamp">
        <af:panelGroup layout="horizontal">
            <af:objectSpacer width="50" height="10"/>
            <af:goLink destination="#{book.homepage}" text="Homepage"/>
        </af:panelGroup>
    </f:facet>
</af:table>
```

```
<af:inputText value="#{book.isbn}" label="ISBN"/>
</af:panelGroup>
</f:facet>
</af:table>
```



Afbeelding 5. Overzicht van de ADF Faces componenten in JDeveloper 10.1.3.2.

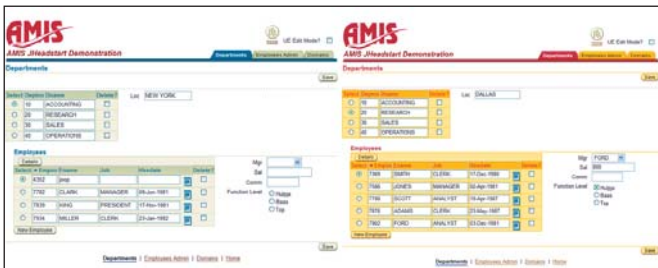


Afbeelding 6. ADF Componenten in actie: Tree, Table, tabbedPanels.

Skins

Hoewel voor websites en op consumenten gerichte webapplicaties een sexy look & feel van groot belang is, wordt ADF Faces over het algemeen ingezet voor de ontwikkeling van applicaties die niet zozeer een marketingdoel nastreven, maar die een administratief proces ondersteunen. De look & feel is daarbij uiteraard ook van belang, maar veel meer vanuit oogpunt van productiviteit en kwaliteit. Zoals eerder beschreven ligt aan de look & feel van ADF Faces applicaties en componenten uitvoerig onderzoek ten grondslag. Over structuur en productiviteit voor de eindgebruiker valt ook weinig te klagen. Wel klinkt regelmatig enig gemopper over de standaardkleuren en inrichting van de schermen, vooral omdat het zo voorspelbaar wordt op een gegeven moment - voor veel omgevingen juist een voordeel.

Het is belangrijk te weten dat ADF Faces niet zomaar een look & feel implementeert: het onderliggende stylesheet-mechanisme heeft ingebouwde ondersteuning voor alle courante en minder courante browsers, browsersversies of besturingssystemen. Uitzonderingen en bekende bugs voor speciale versies van speciale browser in speciale talen zijn ingebouwd. Multi-browser support voor een ADF Faces applicatie komt met het framework mee! Dat bespaart uiteraard aanzienlijk op kosten en frustratie tijdens testen en ontwikkeling van de applicatie. Het blijft echter betrekkelijk eenvoudig om look & feel van ADF Faces applicaties te beïnvloeden met behoud van de multi-browser support. Door eenvoudige manipulatie met een stylesheet kunnen de kleuren en fonts van de applicatie worden beïnvloed. Uiteraard kunnen eigen logo's en schermteksten worden gedefinieerd.

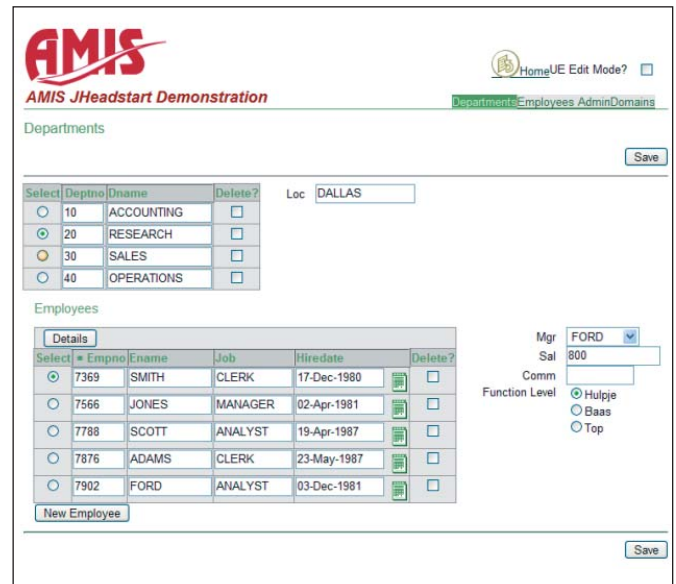


Afbeelding 7. Dezelfde pagina, links met de standaard BLAF skin en rechts met een licht aangepaste variant.

Een meer ingrijpende aanpassing van het uiterlijk van de applicatie kan worden gerealiseerd met een zogeheten *skin*. Een skin is een stylesheet, ontwikkeld binnen bepaalde richtlijnen, die aanpassing van alle aspecten van de weergave van de ADF Faces componenten mogelijk maakt. Een skin bouwt voort op de *simple skin* die Oracle meeleverd. Bovenstaande applicatie ziet er met deze simple skin uit als in afbeelding 8, en kan vanaf dat punt verder opgetuigd worden.

Skins kunnen dynamisch gezet worden. Dat betekent dat voor

iedere gebruiker een andere skin kan worden toegepast, afhankelijk van persoonlijke voorkeur of bijvoorbeeld de afdeling of organisatie waarvoor de gebruiker werkzaam is. Ook zou een gebruiker 'on the fly' een skin kunnen selecteren.



Afbeelding 8. Dezelfde pagina als in de vorige afbeelding, nu met de Simple Skin. Dit is de beste basis voor ingrijpend maatwerk aan de ADF Faces look & feel.

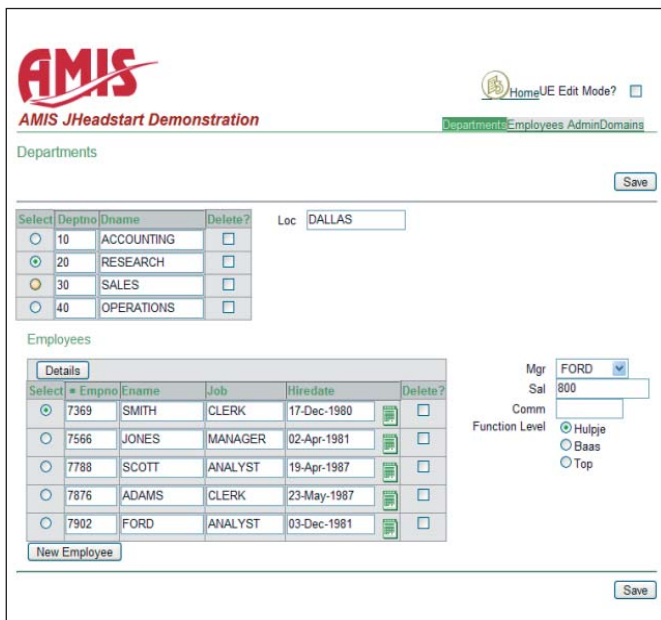
Client Side validatie

JSF kent tussen de standaardcomponent zogeheten Validators. Deze zorgen voor validatie van de waarde die door gebruiker is ingevuld in een bepaalde component. Deze validatie wordt aan de server-kant geplaatst, dus pas na submitten van de pagina. ADF Faces biedt zijn eigen, uitgebreide set van Validator-componenten, die ook Client Side validatie kunnen doen. De componenten genereren de daarvoor benodigde JavaScript code die ervoor zorgt dat als een gebruiker een waarde invoert in een veld en het veld verlaat, zonder browser/servercommunicatie direct validatie wordt uitgevoerd en zondig een foutmelding wordt getoond.

Rijke gebruikerservaring door Ajax

De belangrijkste gedachte achter de Ajax-hype is de dynamische verversing van delen van een webpagina als gevolg van asynchrone – dat wil zeggen: voor de gebruiker niet zichtbare – communicatie tussen browser en applicatieserver. Hiermee kunnen ondermeer instant veld-validaties worden gedaan, suggesties worden geboden in een drop-down list naar aanleiding van de eerste letters, en delen van een scherm verborgen of juist getoond worden naar aanleiding van het aanvinken van een checkbox of invullen van een waarde. In het algemeen kan er met Ajax heel actief en dynamisch worden gereageerd op acties van de eindgebruiker zonder deze in zijn handelen wordt

bepikt door zandlopers of een niet reagerende browser. Binnen ADF-Faces wordt deze aanpak Partial Page Rendering (PPR) genoemd, geïmplementeerd op een wat alternatieve manier die ook stokoude browsers versies ondersteunt en anderszins extra functionaliteit biedt. In het kort kan bij iedere component op een pagina worden aangegeven – declaratief, zonder coderen – dat een wijziging in de waarde van de component moet leiden tot een asynchrone communicatie van de browser met de server. De server verwerkt de gewijzigde waarde(n) en stuurt eventueel verse HTML naar de browser. De ADF Faces JavaScript engine verwerkt de browser response en richt zich op elementen in de pagina waarvan door de ontwikkelaar was aangegeven dat ze verversst moeten worden.



Afbeelding 9. Voorbeeld van AJAX met ADF Faces - Alle elementen in deze applicatie kunnen Partial Page Refresh activeren.

Bovenstaande afbeelding beschrijft een applicatie (de broncode van deze applicatie kan in [7] worden gedownload) waarmee theaterkaartjes kunnen worden besteld. Alle elementen op deze pagina kunnen PPR activeren die leidt tot verversing van één of meer andere elementen. Als bijvoorbeeld de selectie van 'Category of Shows' wordt gewijzigd wordt direct de Show lijst aangepast. Als een andere Show wordt gekozen wordt de getoonde afbeelding gesynchroniseerd, wordt de gekozen datum gevalideerd, wordt de totaalprijs opnieuw berekend, wordt gevalideerd of het aantal gewenste kaartjes wel beschikbaar is voor de voorstelling en wordt het Seating Arrangement element aan- of uitgezet omdat de ene voorstelling wel en de andere niet een speciale stoelindeling hanteert. Deze dynamische aanpassing wordt declaratief gerealiseerd. Neem bijvoorbeeld het invoerveld voor het aantal kaartjes. Hier is het attribute autoSubmit op true gezet. Dat geeft aan:

als de waarde in dit veld wordt veranderd door de eindgebruiker, start een PPR:

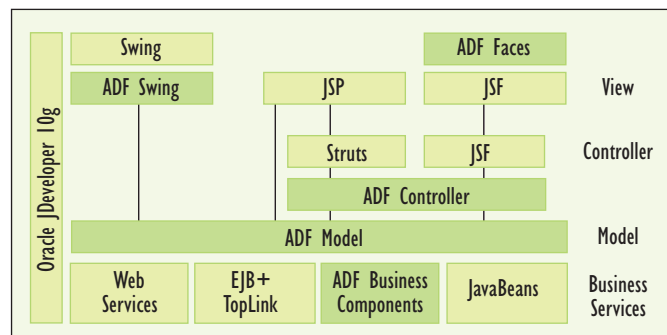
```
<af:inputText value="#{bookingBean.partyCount}"
autoSubmit="true" id="pc" >
```

Het corresponderende veld Total Price: hier geeft het attribute partialTriggers aan dat als er een PPR wordt uitgevoerd als gevolg van het fl:pc element (het veld voor het aantal kaartjes) – of voor de keuze van de voorstelling of de stoelcategorie – dat dan dit veld moet worden verversst.

```
<af:outputText value="#{bookingBean.totalPrice}"
partialTriggers="fl:pc fl:show fl:seating" />
```

Data Intensieve applicaties met ADF Data Binding

ADF Faces wordt in veel gevallen ingezet voor de ontwikkeling van data-intensieve webapplicaties. Daarbij wordt veelal gebruik gemaakt van ADF Model – de Data Binding op basis van JSR-227. Hoewel ADF Model niet de enige manier is vanuit ADF Faces om data uit achterliggende enterprise services te gebruiken, is het wel een heel eenvoudige, door de vergaande integratie binnen de IDE. De aansluiting op databronnen van de ADF Faces componenten wordt grotendeels gerealiseerd met een 'sleur & pleur' (drag & drop) operatie.



Afbeelding 10. De ADF architectuur.

Alvorens ADF Faces componenten 'data bound' te maken, moeten de ADF Model Data Controls worden gepubliceerd. Als gebruik wordt gemaakt van ADF Business Components (voorheen BC4j) om data-opvraag & manipulatie met de relationele database te doen, worden de applicatiemodules automatisch als Data Controls beschikbaar gesteld en zijn de view-objecten binnen de Application Module beschikbaar als datacollecties om tabellen, trees en forms op te baseren. Ook andere dataleveranciers kunnen worden gebruikt als Data Control, zoals webservices, files, URL's of Java Beans, zoals door het Spring Framework ondersteunde Data Access Objects of op Hibernate gebaseerde dataleveranciers.

OraVision bouwt Oracle-oplossingen waarin documenten, transacties en bestaande systemen samenwerken.
OraVision staat bekend als *the mid-office company*.
OraVision bouwt vanuit haar geheel eigen visie: kwaliteit staat centraal.



Kwaliteit in kennis

OraVision beschikt over enorme ervaring in Oracle-, Java- en integratietechnologieën. Bij ons staat de techniek echter nooit op zichzelf. Juist bij mid-office en document-integratie toepassingen laten we de technologie tot volle bloei komen.

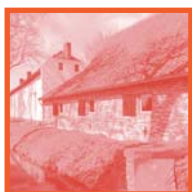
Kwaliteit in werk

Klanten geven OraVision al jaren het vertrouwen om geavanceerde ICT-toepassingen te realiseren die tegelijk gebruikersvriendelijk zijn. Onze mid-office oplossingen bevinden zich immers in het hart van elke bedrijfsvoering.

Kwaliteit in samenwerking

Bij OraVision staat niet alleen technische kwaliteit hoog in het vaandel, ook onze stijl is onderscheidend. Vanuit onze Limburgse basis investeren we nadrukkelijk in persoonlijke relaties en genieten van het goede leven.

Geïnteresseerd in de visie van OraVision op Oracle, Java, integratie en mid-office? Bezoek www.oravision.com en abonneer u gratis op de OraVisionair.



DE ZEKERHEID VAN EEN VASTE BAAN MET DE VOORDELEN VAN ZELFSTANDIG ZIJN

Maak kennis met de Fixion formule.

Bij ons kun je kiezen voor een salaris op maat. Veel winstdeling of juist een hoog basissalaris? Bepaal zelf de beloningstructuur die bij jou past!

Fixion is een jonge, ambitieuze organisatie met een kern van ervaren medewerkers, gespecialiseerd in Oracle en Java technologie. Onze specialisten opereren landelijk voor zowel eindgebruikers als derde partijen. Wij zoeken ervaren professionals die ondernemend zijn, willen delen in de resultaten maar ook meedenken en beslissen over hoe die gehaald worden.

Fixion b.v. | Science Park Eindhoven 5242 | 5692 EG Son
Tel. 040 29 00 632 | www.fixion.nl



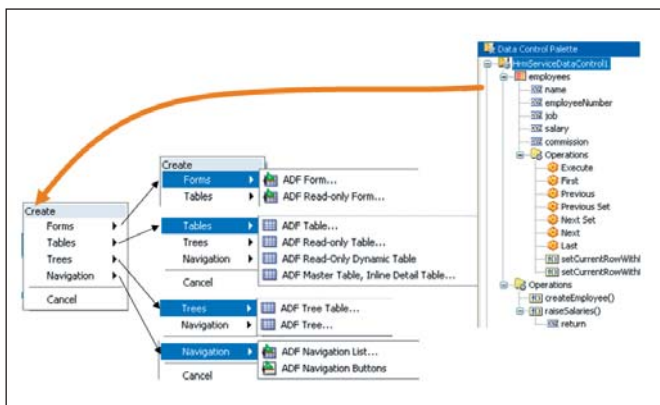
OPTIMIZE

Mededeling voor adverteerders

Optimize is het eerste onafhankelijke tijdschrift over Oracle-databases en de tools die daarop draaien. Optimize heeft een scherp oog voor techniek en de marktontwikkelingen rondom Oracle, haar partners en concurrenten. Optimize biedt praktische en professionele informatie voor IT-managers, ontwikkelaars, database-beheerders en projectmanagers, geschreven door en bestemd voor Oracle-professionals. U kunt deze doelgroep in één keer bereiken door in Optimize te adverteren.

Nummer 3 verschijnt op 14 juni.
De sluitingsdatum voor adverteerders is 22 mei.

De Data Controls die in de applicatie kunnen worden gebruikt, zijn gepresenteerd in het Data Control Palette. We zien de Data Controls en al hun collecties en operaties en daarbinnen weer alle beschikbare attributen, parameters en resultaten. Elk element dat in het Data Control Palette wordt getoond, kan met Drag & Drop naar onze JSF pagina worden gesleept. Als het gesleepte element wordt losgelaten, opent een dialoog die, afhankelijk van het type data element, een keuze aanbiedt uit de verschillende componenten die relevant zijn. Een Collectie kan bijvoorbeeld als tabel, tree, navigatiebuttons en meer worden gepresenteerd.



Afbeelding 11. Drag & Drop Data Control elementen als Data Bound ADF Faces componenten op een pagina.

Als er in de pagina ADF Faces componenten staan die via ADF Data Binding op achterliggende Data Controls zijn aangesloten, wordt voor de pagina een zogenaamde PageDefinition file aangemaakt. Hierin is vastgelegd welke data-elementen voor de pagina worden gebruikt en hoe deze binnen JSF beschikbaar zijn. ADF Faces voegt functionaliteit toe – onder meer een ServletFilter – aan de webapplicatie om te zorgen dat de data-elementen zoals die zijn gespecificeerd in de PageDefinition op het juiste moment geïnitieerd worden. Vanuit de JSF-componenten kan vervolgens via de gebruikelijke EL-expressies van deze data-elementen gebruik worden gemaakt, bijvoorbeeld voor het Ename-veld in een pagina op de EMP tabel:

```
<af:inputText value="#{bindings.EmpView1Ename.inputValue}"
  label="#{bindings.EmpView1Ename.label}"
  required="#{bindings.EmpView1Ename.mandatory}"
  columns="#{bindings.EmpView1Ename.displayWidth}">
  <af:validator binding="#{bindings.EmpView1Ename.validator}"/>
</af:inputText>
```

De data binding wordt ondersteund door dit (gegenereerde) fragment uit de PageDefinition file:

```
<pageDefinition>
  <executables>
    <iterator id="EmpView1Iterator" RangeSize="10" Binds="EmpView1"
      DataControl="HrmServiceDataControl"/>
  </executables>
  <bindings>
    <attributeValues id="EmpView1Ename" IterBinding="EmpView1Iterator">
      <AttrNames>
        <Item Value="Ename"/>
      </AttrNames>
    </attributeValues>
  </bindings>
</pageDefinition>
```

De elementen uit de PageDefinition file worden door het ADF framework onder de key `#{bindings}` aan JSF-componenten ter beschikking gesteld. Binnen de PageDefinition vind je de koppeling van het attributeValues element – waar het ADF Faces InputText component op is gebaseerd – aan de iterator (een generieke representatie van een data collectie), en de koppeling van de iterator aan het EmpView1 ViewObject. Dit is voor het framework voldoende informatie om in het InputText element de juiste waarde te tonen – de Ename van de huidige Employee – en ook wijzigingen in die naam op de juiste manier terug te sturen naar het ViewObject en indirect de database.

Merk op dat dit datagebonden inputveld geen enkele kennis heeft van de database of zelfs maar het ADF BC ViewObject. Het ADF Model schermt de afkomst van het data element volledig af. Als we ooit besluiten om Employees uit een webservice te halen in plaats van via een ADF BC ViewObject uit een relationele database, heeft dat geen impact om de ADF Faces webapplicatie. Bovendien hebben we dit veld met data binding kunnen aanmaken met niets meer dan een drag & drop operatie! Het is overigens ook mogelijk om andere JSF-componenten dan die van ADF Faces data bound te maken.

Een nog hogere productiviteit in de ontwikkeling van data-intensieve webapplicaties op basis van de ADF Faces kan worden bereikt door gebruik te maken van JHeadstart, een extra design time plugin voor JDeveloper. JHeadstart genereert zowel de ADF Model PageDefinition files als de JSF pagina's op basis van een meta-definitie van de applicatie en de schermen.

Conclusie

ADF Faces is 'gewoon' een JSF-implementatie, die voldoet aan alle standaarden en kan worden uitgerold op iedere applicatieserver. ADF Faces is ook een heel volwassen, complete en rijke verzameling componenten die erg goed geïntegreerd is met de JDeveloper IDE en daarmee goede ondersteuning biedt voor het zeer productief ontwikkelen van JSF-applicaties. De mogelijkheid om Data Binding te benutten vergroot de kracht van het framework in belangrijke mate.

Door de wijdverspreide inzet van ADF Faces en de sterke

commitment vanuit Oracle – ADF (Faces) is immers cruciaal voor Oracle (Fusion) Applications – is de toekomstvastheid van ADF Faces erg groot, getuige ook de aanstormende Rich Components in release 11g.

Referenties

- 1 The Complete Reference: Java Server Faces – Chris Schalk & Ed Burns, McGrawHill/Osborne, 2006 ISBN 0072262400
- 2 Oracle JDeveloper 10g for Forms & PL/SQL Developers: A Guide to Web Development with Oracle ADF - Peter Koletzke & Duncan Mills, Osborne Oracle Press, 2006 ISBN 0072259604
- 3 OTN Homepage voor ADF Faces: <http://www.oracle.com/technology/products/jdev/htdocs/partners/addins/exchange/jsf/index.html>
- 4 Oracle Browser Look & Feel Guidelines (BLAF) homepage: <http://www.oracle.com/technology/tech/blaf/index.html>
- 5 AMIS Technology Weblog – <http://technology.amis.nl/blog-tientallen-artikelen-en-voorbeelden-van-jsf>
- 6 Workshop Pretty Java Server Faces – een introductie op JSF (Nederlandstalig) <http://www.amis.nl/activiteiten.php?id=316>
- 7 Sources voor dit artikel (JDeveloper 10.1.3.2 Application en Projecten): <http://technology.amis.nl/blog/?p=1652>
- 8 ADF Faces alias MyFaces Trinidad website: <http://incubator.apache.org/projects/adffaces.html> (zie ook: <http://myfaces.apache.org/>)

Lucas Jellema is Technology Manager bij AMIS. Daarnaast schrijft hij artikelen op de AMIS Weblog (<http://technology.amis.nl/blog>) en in tijdschriften als Optimize, Java Magazine en Java Developer Journal. Hij is zowel Oracle ACE als Oracle Regional Director for Fusion Middleware. Lucas kan bereikt worden via email: jellema@amis.nl.



Oracle lanceert Oracle TimesTen In-Memory Database 7

Oracle maakt de beschikbaarheid bekend van Oracle TimesTen In-Memory Database Release 7. Dit product biedt betere integratie en caching met Oracle Database 10g voor het meten van instant response tijden en high-throughput requirements van real-time enterprises. Oracle TimesTen wordt gebruikt in verschillende sectoren zoals financiële dienstverlening, telecommunicatie en defensie, waar real-time applicaties essentieel zijn voor succes van de organisatie. Oracle TimesTen 7 levert dynamische database caching van Oracle Databases voor performance-kritische enterprise applicaties zoals transactie processing van grote hoeveelheden data, business activity monitoring, real-time business intelligence, CRM en customer self-service. Nieuwe, verbeterde features in Oracle TimesTen 7 zijn onder meer:

- Data type compatibiliteit - vertrouwde Oracle Database 10g data typen zijn nu beschikbaar in Oracle TimesTen 7 voor eenvoudiger applicatie-ontwikkeling en het cachen van data in memory;
- Nieuwe SQL features - Oracle TimesTen 7 bevat verbeterde SQL-functionaliteit, met vergelijkbare semantiek en gedragingen als Oracle Database 10g;
- Globaliseringsfunctionaliteit - meer dan 50 database karaktersets en 80 talen worden nu ondersteund in Oracle TimesTen 7;
- Automatische data aging en on-demand data loading - data in Oracle TimesTen 7, gebruikt in cache, kunnen vanuit het geheugen dynamisch geladen worden en voorzien van een time stamp;
- Cross-tier high availability - data kunnen automatisch gesynchroniseerd worden over gerepliceerde Oracle TimesTen databases en Oracle Databases.

Oracle TimesTen 7 ondersteunt ook Oracle RAC, Oracle Fusion Middleware, Oracle SQL Developer en Oracle JDeveloper.

Oracle TimesTen In-Memory Database is verkrijgbaar vanaf 12.000 US dollar per CPU, afhankelijk van de omvang van de in-memory database. Twee opties, Replication (TimesTen naar TimesTen) en Cache Connect to Oracle, zijn verkrijgbaar vanaf 6.000 US dollar per CPU, afhankelijk van de omvang van de in-memory database. De software is te downloaden op <http://www.oracle.com/technology/software/products/timesten>.