

JHeadstart 10.1.3 voor ADF Faces

Een grote sprong vooruit

Half augustus publiceerde het nog steeds in Nederland gevestigde ontwikkelteam de 10.1.3 Release van JHeadstart, met ondersteuning voor JDeveloper 10.1.3 en ADF Faces. Deze nieuwste release, de vierde major release in het bestaan van het product sinds 2001, betekent een enorme sprong vooruit. De meest opvallende ontwikkelingen in deze release betreffen naast de generatie van ADF Faces – Oracle's implementatie van Java Server Faces –, de aanzienlijk verbeterde JDeveloper plugins die het leven van de ontwikkelaar veel gemakkelijker maken, het generator template mechanisme dat een veel groter percentage van de applicatie genereerbaar maakt en een reeks nieuwe generatie opties zoals Table Overflow, Unbound Items en Deep Linking navigatie. Daartegenover staan het niet langer genereren van (pure) JSP en UNIX pagina's.

JHeadstart is in de afgelopen jaargangen van de Optimize herhaaldelijk onder de loep genomen. Ondertussen is JHeadstart een product dat wereldwijd door honderden organisaties wordt toegepast en dat een van de meest actieve OTN Discussie Forums kent - <http://forums.oracle.com/forums/forum.jspa?forumID=38>. JHeadstart blijkt vaak een middel te zijn om de als enorm ervaren drempel tussen klassiek Oracle Forms ontwikkeling en ADF te overwinnen.

Men zou zich kunnen afvragen: wat is JHeadstart nu eigenlijk? Is het:

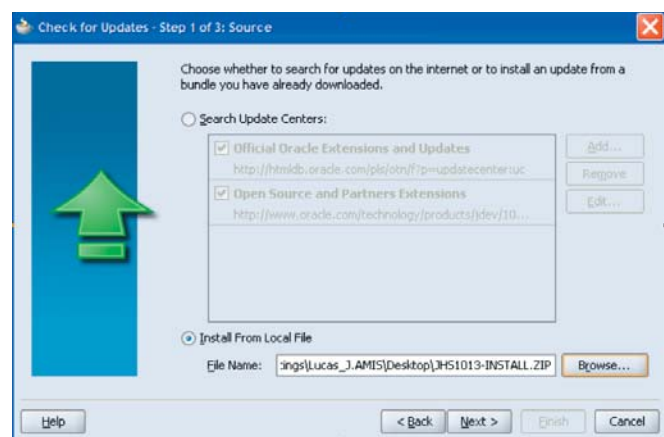
- Een generator die op grond van declaratieve applicatiedefinities een ADF-applicatie genereert
- Een tool dat bestaande Forms Modules uit Oracle Designer migreert naar de ADF Technology Stack
- Een verzameling Best Practices voor het gebruik van ADF Technologie bij de ontwikkeling van webapplicaties
- Een productivity booster die in extreem korte tijd het fundament – JSF pagina's en configuratie files, resource bundles, ADF Page Definitions – van een ADF Applicatie ophoest

- Een leer-curve-afvlakker waarmee Oracle-ontwikkelaars zonder noemenswaardige Java kennis – maar met bijvoorbeeld een Oracle Designer of Forms achtergrond – in korte tijd productief kunnen meedraaien in ADF ontwikkeltrajecten.

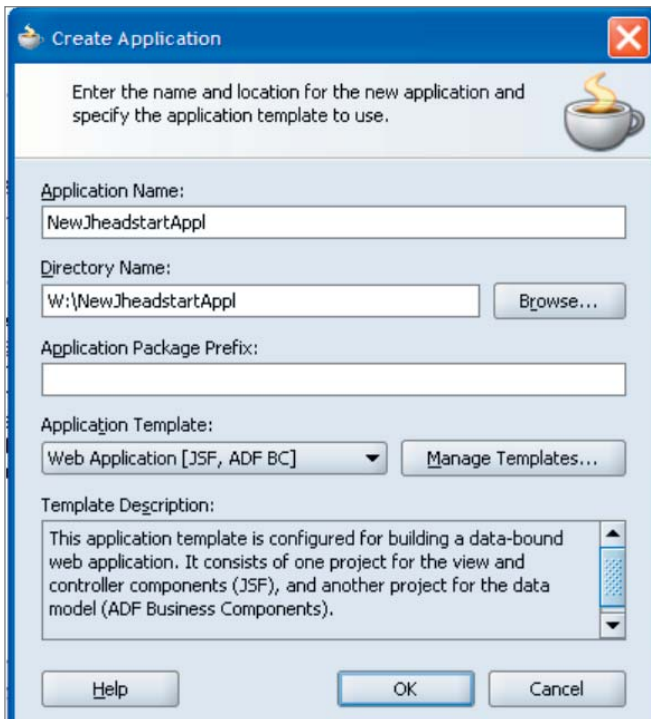
Het simpele antwoord is: JHeadstart is dat allemaal. Afhankelijk van de situatie kan JHeadstart al deze gedaantes aannemen. Zelfs al zou je bijvoorbeeld geen enkele pagina in je ADF-applicatie genereren met JHeadstart, dan nog zou de aanschaf van een enkele licentie de moeite waard zijn, door de schat aan ervaringen en best practices met ADF die in JHeadstart zijn vastgelegd en gedemonstreerd. Door de code van JHeadstart te bekijken, krijg je veel inzicht in de interne werking van ADF Faces en ADF Binding. De leercurve voor de ADF Technology Stack is aan de steile kant, maar JHeadstart vlakt deze flink af.

ADF Faces

Laten we eens kijken hoe je een ADF-applicatie ontwikkelt met JHeadstart. Voor het opzetten van de ontwikkelomgeving hebben we JDeveloper 10.1.3 nodig, toegang tot een (bij voorkeur Oracle) database en de JHeadstart 10.1.3 zip-file. De installatie van JHeadstart wordt gedaan door de zipfile te selecteren van-



Figuur 1. Installatie JHeadstart vanuit het Check for Update Center in JDeveloper 10.1.3



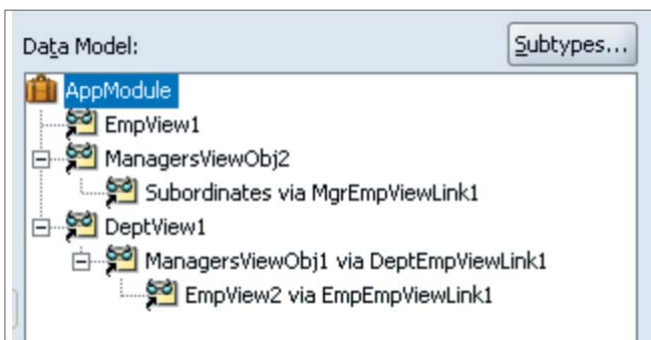
Figuur 2. Creëer een nieuwe Application

uit het Check for Updates mechanisme. Binnen JDeveloper maken we een New Application met het Web Application [ADF BC, JSF] Technology Template.

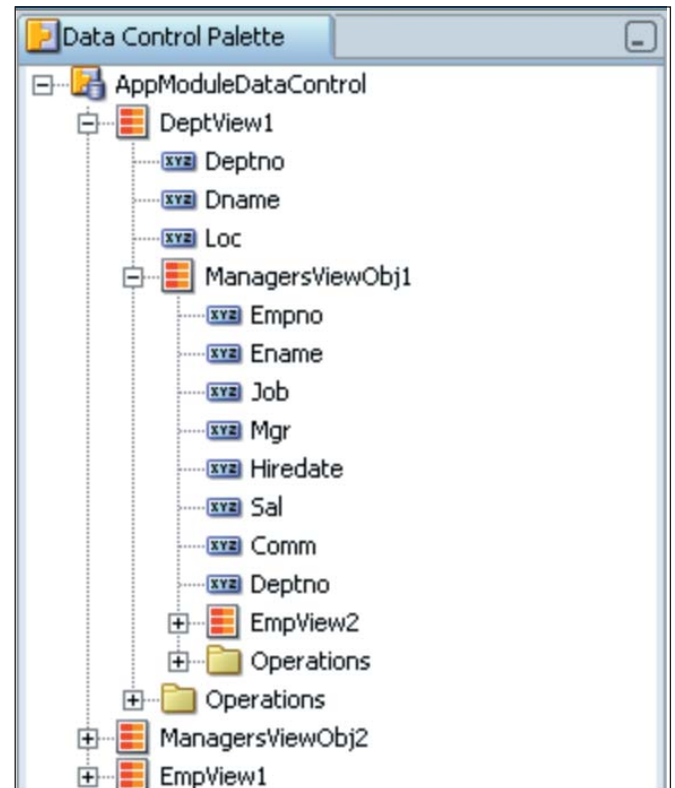
Inrichting van de Data Services

In het Model-project gaan we vervolgens de Business Components creëren die voor onze applicatie noodzakelijk zijn. Deze stap is vergelijkbaar met de ontwikkeling van specialistische Views voor Module Components in Oracle Designer.

De ViewObjecten die we uiteindelijk opnemen in de Application Module en hun onderlinge ViewLinks bepalen welke data services aan de applicatie beschikbaar worden gesteld. Ze worden automatisch gepubliceerd als ADF DataControls en komen daarmee beschikbaar voor JHeadstart.



Figuur 3a. Data Model van de ADF BC Application Module.



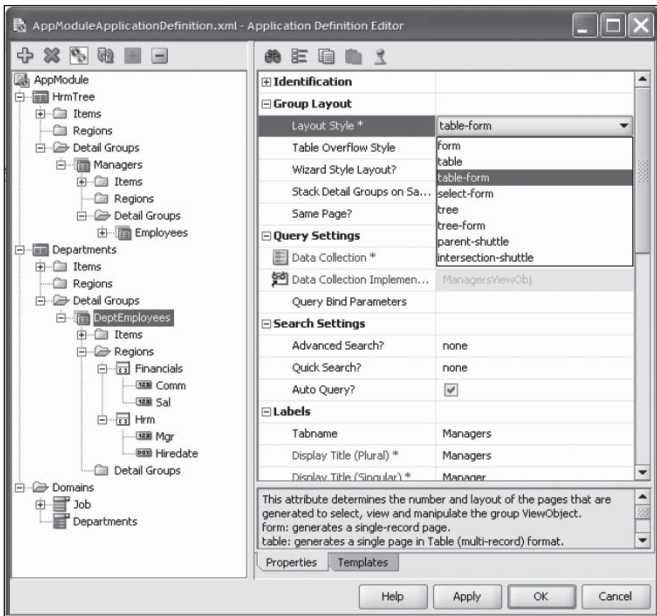
Figuur 3b. ADF Binding Data Control Palette.

Ontwikkeling van de webapplicatie

Nu het fundament voor de applicatie is gelegd kunnen we ons richten op de webapplicatie zelf, het ViewController project. Allereerst gaan we dit JHeadstart enablen – door de gelijknamige optie in het RMB (right mouse button) Menu op het ViewController project te kiezen. Vervolgens kiezen we uit datzelfde menu de optie New JHeadstart Application Definition. Dit maakt voor ons een basis applicatiedefinitie op grond van de Data Control (en indirect dus de Application Module) en al zijn geneste Collections.

Dit is het moment waarop we onze ADF Web Applicatie declaratief gaan definiëren met de nieuwe JHeadstart Application Definition Editor. De afzonderlijke JHeadstart ViewObject en EntityObject editors zijn vervallen in de 10.1.3 versie en ook het handmatig editen van de domains definitie-file behoort tot het verleden. Alle functionaliteit met betrekking tot de applicatie, de data usages, display eigenschappen, domeinen van toegestane waarden en navigatie structuren zijn terug te vinden in één tool (zie figuur 4).

De Application Definition bevat Groups. Een Group correspondeert met een ViewObject Usage en leidt tot een of meer drie pagina's. Groups kunnen detailgroups hebben – voor Master Detail schermen en Tree componenten. Groups zijn opgebouwd uit Items – meestal gebaseerd op Attributen uit de

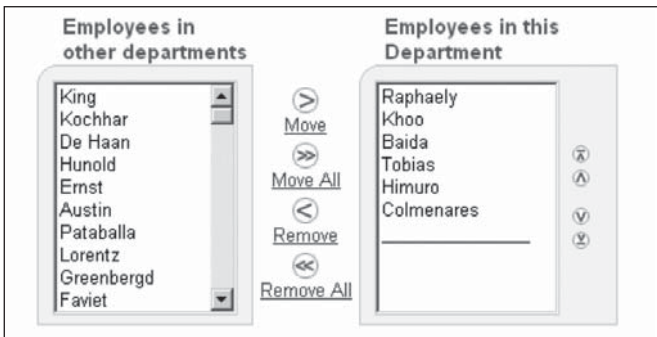


Figuur 4. De nieuwe JHeadstart Application Definition Editor.

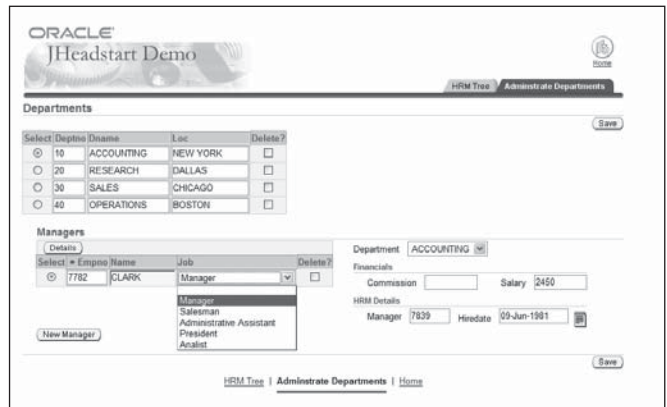
onderliggende Collection, en momenteel dus ook indirect op ViewObject Attributes.

Zoals in Oracle Designer met Item Groups verschillende onderling verwante items binnen Item Groups kunnen worden samengebracht kent JHeadstart de Regions. Met drag & drop kunnen items naar Regions worden gesleept. Regions zijn op het scherm herkenbaar door een kader, een eigen titel en indentatie. De 10.1.3 release ondersteunt stacked Regions – met een tabblad per Region -, geneste Regions en ook Regions met Detail Groups. Deze kunnen binnen een Group gecombineerd worden.

De belangrijkste eigenschap van een Group is de layout style. Deze bepaalt voor een belangrijk deel het uiterlijk en de interactie van de eindgebruiker met de Group in de browser. De meestgebruikte layout-styles zijn Form en Table – voor Single en Multi-record. Andere mogelijkheden zijn Table-Form (waarbij in een Tabel-layout de belangrijkste attributen van de rijen worden getoond en op een tweede pagina met Form-layout van



Figuur 5. De Shuttle Component voor het toewijzen van Details aan een Master.

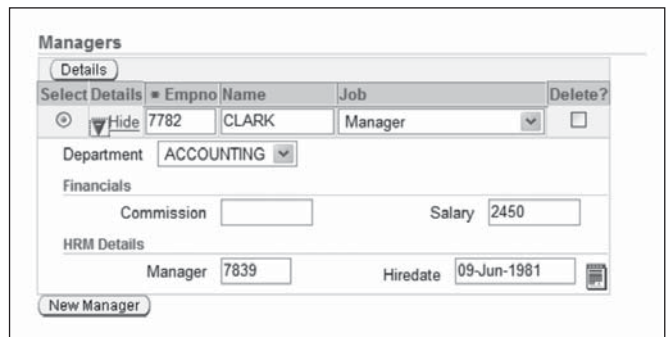


Figuur 6. Gegeneerde applicatie met Master-Detail, Table Overflow Style Right.

een geselecteerd record alle waarden worden getoond), Tree en Tree Form voor een hiërarchische datastructuur, al dan niet met Edit-mogelijkheid voor de geselecteerde node en Shuttle waarmee een Group als een Shuttle element wordt gegeneerd waarmee gebruikers Details aan een Master kunnen toewijzen of Intersection records kunnen aanmaken.

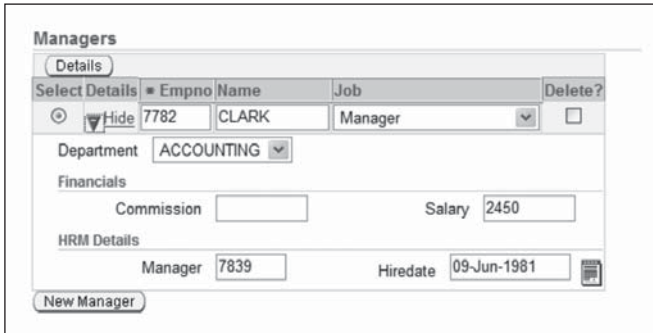
Generatie van de applicatie is snel: voor de voorbeeldapplicatie uit dit artikel was minder dan drie seconden benodigd. De generatie kan overigens voor onderdelen van de applicatie worden gedaan en verder kan een project meerdere applicatie-definitie files bevatten waarvan de generatieresultaten worden samengevoegd. De gegeneerde applicatie voor de Departments Group ziet er als in Figuur 6.

De DeptEmployees Detail Group had het property Same Page op true staan en is dus op dezelfde pagina beland. Beide groepen hebben Table als lay-out stijl, ook al heeft Department 10 maar één Manager. We zien duidelijk de twee regions – Financials en HRM Details. We zien ook een ander nieuw JHeadstart 10.1.3 feature – de Table Overflow Style. Ontwikkelaars die met Designer hebben gewerkt zullen nostalgische gevoelens krijgen naar een vergelijkbaar property in Module Componenten. De overflow stijl geeft aan waar de



Figuur 7. Table Overflow Style in-line.

items die niet in de tabel zelf komen te staan moeten worden geplaatst op de pagina. JHeadstart kan drie overflow stijlen genereren: right, below en in-line. In-line bestaat niet in Oracle Designer; deze stijl resulteert in een zogenaamd Detail Disclosure element binnen de tabel zelf:

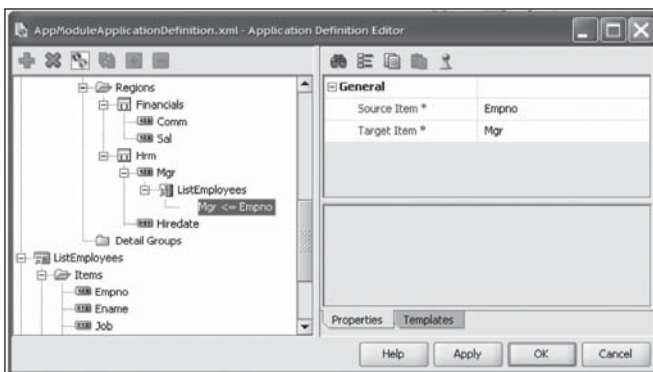


Figuur 7. Table Overflow Style in-line.

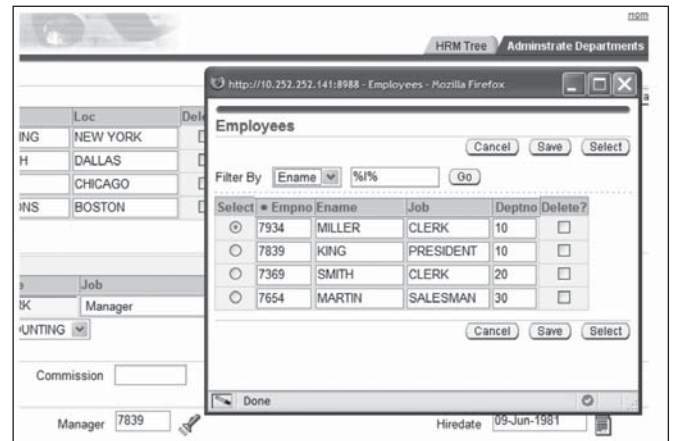
Of JHeadstart ook ooit de Designer Form Generator overflow stijl Spreadtable zal ondersteunen is onduidelijk maar niet waarschijnlijk. Een belangrijk onderdeel van data georiënteerde webapplicaties zoals we die met ADF ontwikkelen wordt gevormd door de DropDown lijstjes en de Lists of Values. Een Employee wordt toegewezen aan een Department en krijgt een Job uit een lijstje van toegestane waarden.

JHeadstart kent twee types domeinen: statisch – met een hard gecodeerd lijstje waarden – en dynamisch – een lijst die dynamisch wordt opgebouwd met behulp van een ADF Data Collection – indirect een ViewObject. Een Item kan ondermeer de Display Data Type waarden DropDownList, Lov, Radio Horizontal en RadionVertical hebben. In al deze gevallen wordt het item gekoppeld aan een domein dat de waarden waaruit de gebruiker voor dat Item kan kiezen worden onttrokken.

Voor een meer geavanceerde List of Values kunnen we een speciale Group ontwikkelen met de door ons gewenste velden en teksten. Deze Group kunnen we vervolgens verbinden aan een Item met display type is Lov. De gegeneerde applicatie laat



Figuur 8. Definitie van een List of Values.

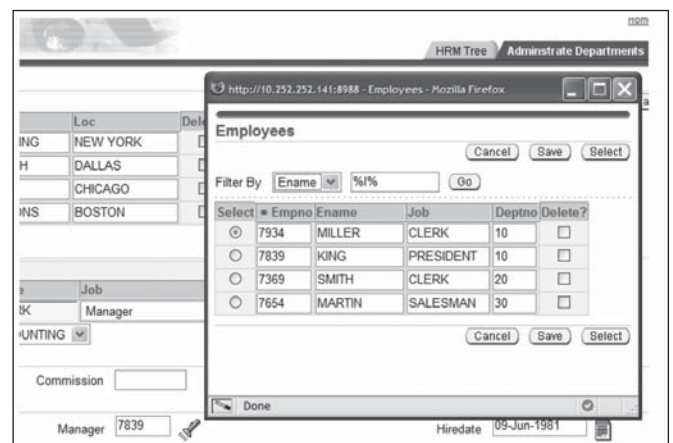


Figuur 9. De gegeneerde List of Values in actie.

de gebruiker via een Lov button een popup scherm openen – vergelijkbaar bijvoorbeeld met LOV's in Oracle Forms - waarin de gebruiker een record kan zoeken en selecteren.

Geavanceerde functionaliteit

Met de ondersteuning van ADF Faces is ook de ondersteuning van EL expressies voor het dynamisch afleiden van allerlei eigenschappen verder toegenomen. JHeadstart stelt ons in staat om ondermeer prompt, tooltip, grootte, default waarde, al dan niet verplicht, disabled en zelfs zichtbaar zijn, waarden van Query Bind Parameters door middel van EL-expressies te definiëren en daarmee dynamisch afleidbaar te maken. Het veld COMM bijvoorbeeld kan eenvoudig disabled of zelfs onzichtbaar worden gemaakt als het veld JOB niet de waarde SALESMAN bevat:



Figuur 10. Specificeer het dynamisch disablen/enablen van Commission afhankelijk van de waarde van Job.

Een ander nieuw en krachtig mechanisme is de Deep Link. Deze stelt ons in staat om van een record op de ene pagina naar een verwant gegeven op een andere pagina te navigeren,

bijvoorbeeld van een Employee naar een pagina met gegevens over zijn Manager. De ondersteuning voor role-based security, de generatie van wizards, ondersteuning van File Upload/Download, de uitgebreide ondersteuning voor geavanceerde zoek-pagina's: allemaal onderwerpen die meer aandacht verdienen - maar dit artikel kan maar zo lang zijn.

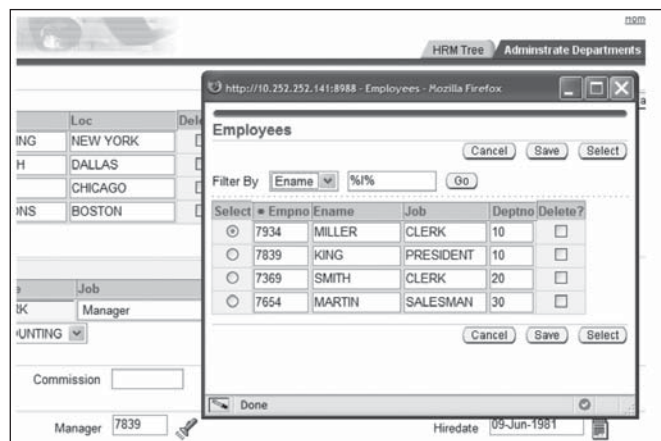
Speciale aandacht in de JHeadstart 10.1.3 Release verdient het Template Mechanisme. Dit mechanisme stelt ons in staat om handmatige, post-generatie aanpassingen van onze applicatie onderdeel te maken van verdere generatie-slagen. Dus ook componenten die na generatie zijn aangepast blijven genereerbaar! Daarnaast kunnen we met het Template Mechanisme voor onze volledige applicatie de functionaliteit van de JHeadstart Applicatie Generator aanpassen.

Het Template Mechanisme is gebaseerd op de Apache Jakarta Velocity engine en de Spring Framework Inversion of Control container. Ieder onderdeel van de applicatie – items, menu, titels en andere schermteksten, tree, shuttle – worden door de JHeadstart generator gegenereerd op grond van een template. Normaal gesproken is dat het meegeleverde default template, maar we kunnen een alternatief template specificeren. In dat zogenaamde Custom Template kunnen we – met gebruikmaking van generator variabelen die verwijzen naar property's uit de Application Definition file – eigen code fragmenten opnemen. Handmatige, post-generatie aanpassingen kunnen we zo in een Custom Template opnemen en de plaats laten innemen van het Default Template voor een bepaald item.

Praktijkervaringen

Sinds juli 2006 zijn we bij AMIS met een team actief met de JHeadstart 10.1.3 release – voor de ontwikkeling van een applicatie voor de grondexploitatie door gemeentes.

Deze applicatie is geheel op ADF gebaseerd, is zeer data-intensief en vereist tegelijk een gebruikersvriendelijke interface voor tamelijk complexe datastructuren en operaties. Aanvankelijk



Figuur 11. 100% Gegeneerde Tree-Form layout stijl in de Pagoni applicatie.

was ons plan JHeadstart alleen in te zetten voor de initiële generatie, om voorbeelden te bieden van veel voorkomende ADF constructies en mogelijk de eenvoudige schermen volledig te genereren. Ondertussen hebben we ondervonden dat de generatie-functionaliteit zo krachtig is dat we vrijwel alle gewenste UI logica toch kunnen genereren en dat we waar dat niet het geval is met het Template Mechanisme op basis van Custom Templates toch de generatie kunnen vasthouden – terwijl dat eigenlijk niet eens ons streven is.

We hebben ondervonden dat JHeadstart ontwikkelaars helpt om veel sneller up and running te zijn met ADF Faces en ADF Binding dan zonder JHeadstart het geval zou zijn, en dat de productiviteitswinst – zelfs voor ervaren ADF-ontwikkelaars – aanzienlijk is.

Conclusie

JHeadstart 10.1.3 biedt krachtige ondersteuning voor de ontwikkeling van ADF Faces applicaties. Op eenvoudige, declaratieve wijze worden applicaties gespecificeerd en vervolgens gegenereerd. JHeadstart stelt ontwikkelaars met weinig tot geen ADF kennis en ervaring snel productief mee te ontwikkelen aan een ADF-applicatie. Ten opzichte van de voorgaande versies van JHeadstart zetten we met deze 10.1.3 release weer een grote stap in termen van gebruiksgemak, productiviteit, genereerbaarheid en aansluiting bij de (Oracle) Java industrie-standaarden.

Meer weten: op onze AMIS Technology weblog (<http://technology.amis.nl/blog>) vind je regelmatig artikelen over onze praktijkervaringen met JHeadstart. Op 14 september organiseert AMIS de Lancering van JHeadstart 10.1.3 met Steven Davelaar; in deze sessie veel demonstraties van belangrijke nieuwe features en een presentatie van praktijkervaringen met ADF Faces en JHeadstart. Daarnaast zijn er het OTN Discussie Forum: <http://forums.oracle.com/forums/forum.jspa?forumID=38>, de JHeadstart Team Blog <http://blogs.oracle.com/jheadstart> en het JHeadstart Product Center op OTN: <http://www.oracle.com/technology/consulting/9iservices/jheadstart.html>. Op korte termijn zal een uitgekilde 10.1.3 release als gratis download op OTN beschikbaar komen.

Lucas Jellema (Oracle ACE) is Technisch Consultant en Expertise Manager bij AMIS Services in Nieuwegein. Hij is een frequent spreker op conferenties en enthousiast schrijver van artikelen en weblog-postings (zie: <http://technology.amis.nl/blog>). Lucas oriënteert zich met name op services (BPEL, ESB) en Java/J2EE technologie.