

Oracle JHeadstart 11g

Genereren van ADF 11g applicaties

Na de komst van Oracle JDeveloper en ADF 11g is er ook een nieuwe release van Oracle JHeadstart beschikbaar. In een eerder artikel heb ik al aangekondigd dieper in te gaan op het gebruik van JHeadstart in combinatie met ADF 11g. De nadruk ligt op enkele nieuwe functionaliteiten, met name het ADF controller framework.

Oracle JHeadstart 11g genereert zeer geavanceerde ADF-applicaties waarbij Rich ADF Faces als View, ADF Task Flows als Controller en ADF Business Components als Business Service laag gebruikt worden. Het genereren geschiedt met de JHeadstart Application Generator, door op declaratieve wijze de applicatie te specificeren, door middel van metadata en gebruikmaking van property editors. JHeadstart implementeert daarbij automatisch een verzameling van ADF 11 best practices gerelateerd aan nieuwe ADF features zoals (un)bounded task flows, page fragement, page templates, (dynamic) regions, XML Menu Model en model-driven LOV's. Ook bevat het de JHeadstart Forms2ADF Generator, waarmee Oracle Forms applicaties direct naar ADF applicaties getransformeerd kunnen worden.

New Features

Aan Oracle JHeadstart 11g zijn naast vele kleine verbeteringen de volgende functionaliteiten toegevoegd:

- New Application Level in Application Definition Editor.
- Generation of (un)Bounded Task Flows.
- Ability to Generate Page Fragments and (Dynamic) Regions.
- Enhanced Menu Generation.
- Support for Page Templates.
- Enhanced Support for Deeplinking.
- Easy Reuse of Groups using a Group Region.
- Support for Model-based LOV's.
- LOV Groups Reusable Across Services.
- Enhanced Template Editing.
- JHeadstart Forms2ADF Generator.
- Support for Stretching Layouts.
- New Table Overflow Styles.

- New Allowable Values for Same Page Display Position.
- New Allowable Values for Region Container Layout.
- Ability to Add Bindings to PageDefinition using Velocity Templates.
- Flex items.
- File up/download.
- XML Reformatting.

Integreren van JHeadstart met JDeveloper 11g

Voor het integreren van JHeadstart met JDeveloper 11g is een zip bestand (bijvoorbeeld JHS11.1.1.0.31-INSTALL.ZIP) nodig. Via de Check for Updates wizard in JDeveloper (zie menu Help | Check for Updates...) kan in het Source scherm gekozen worden voor 'Install from local file' waarbij je het zip bestand kunt selecteren. Na de installatie zal JDeveloper herstarten en pagina 'Oracle JHeadstart 11.1.1 Technical Preview - Documentation Index' worden getoond.

Het Model project

In dit artikel worden voorbeelden gebruikt van schermen, gebaseerd op tabellen van het HR schema. Er is daarvoor een IDE Connection (local_mydb1_hr) aangemaakt naar het HR schema van de database. Daarna is een Fusion Web Application (ADF) aangemaakt (Application1) met een Model en een ViewController project. In de database Navigator is de IDE Connection beschikbaar gemaakt voor de applicatie, door deze naar de applicatie te slepen.

De modellaag (Entity Objects, View Objects en Application Module) is verder aangemaakt met behulp van de 'Business Components from Tables' wizard.

Aanmaken van een Application Definition

Ga in JDeveloper naar het ViewController project en kies onder de rechtermuisknop voor 'Enable JHeadstart on this Project'. Daarna start de wizard en worden er automatisch een aantal aanpassingen gedaan, zoals:

- Het koppelen van extra libraries aan het ViewController project (Updating project settings):

- JHeadstart Runtime 11.1.1
- JHeadstart Application Generator 11.1.1
- Het aanmaken van een Application Definition file in directory `../ViewController/properties`.

Hieronder is de code te zien van de Application Definition (JHeadstartApplicationDefinition.xml):

```
<?xml version="1.0" encoding="windows-1252"?>
<Application name="Application1 - ViewController" asfVersion="1.0"
templatesBaseDir="\templates\" viewPackage="application1.view"
viewType="adfFaces" nlsBundle="application1.view.ApplicationResources"
xmlns="http://www.oracle.com/jheadstart/applicationStructure"/>
```

Aanmaken van een Service Definition

Ga in JDeveloper naar het ViewController project en kies onder de rechtermuisknop voor 'New JHeadstart Service Definition'. Met behulp van deze wizard is een nieuwe JHeadstart Service Definition aangemaakt op basis van een ADF BC DataControl. Hierbij is gekozen voor de in het model aangemaakte applicatie module (AppModule1), en alle overige instellingen zijn standaard gelaten.

Door de keuze voor 'New JHeadstart Service Definition' worden er automatisch een aantal aanpassingen gedaan, zoals:

- Het aanmaken van een Service Definition file in directory `../ViewController/properties`

Hieronder is de code te zien van de Service Definition (AppModule1ServiceDefinition.xml):

```
<?xml version="1.0" encoding="windows-1252"?>
<Service name="AppModule1" description="Service Definition for
AppModule1" menuModelFile="/WEB-INF/menu-appmodule1.xml" controller
ConfigFile="/WEB-INF/adfc-config-appmodule1.xml" dataControl="AppModule1
DataControl" dataControlImpl="application1.model.service.AppModule1"
xmlns="http://www.oracle.com/jheadstart/applicationStructure">
  <Group name="Employees" description=" Employees" ...>
  </Group>
  <Group name="Departments" description=" Departments" ...>
  </Group>
  <Group name="Locations" description=" Locations" ...>
  </Group>
  <Group name="Countries" description=" Countries" ...>
  </Group>
  <Group name="Regions" description=" Regions" " ...>
  </Group>
  <Domains />
  <Templates />
</Service>
```

- Het koppelen van de Service Definition aan de Application Definition.

Hieronder is de code te zien van de aangepaste Application Definition (JHeadstartApplicationDefinition.xml):

```
<?xml version="1.0" encoding="windows-1252"?>
<Application name="Application1 - ViewController" asfVersion="1.0"
templatesBaseDir="\templates\" viewPackage="application1.view"
viewType="adfFaces" nlsBundle="application1.view.ApplicationResources"
xmlns="http://www.oracle.com/jheadstart/applicationStructure" default
GroupRegionAccess="groupUIShell">
  <ServiceDefinitionRef location="AppModule1ServiceDefinition.xml"
```

```
name="AppModule1"/>
</Templates/>
</Application>
```

Application Definition Editor

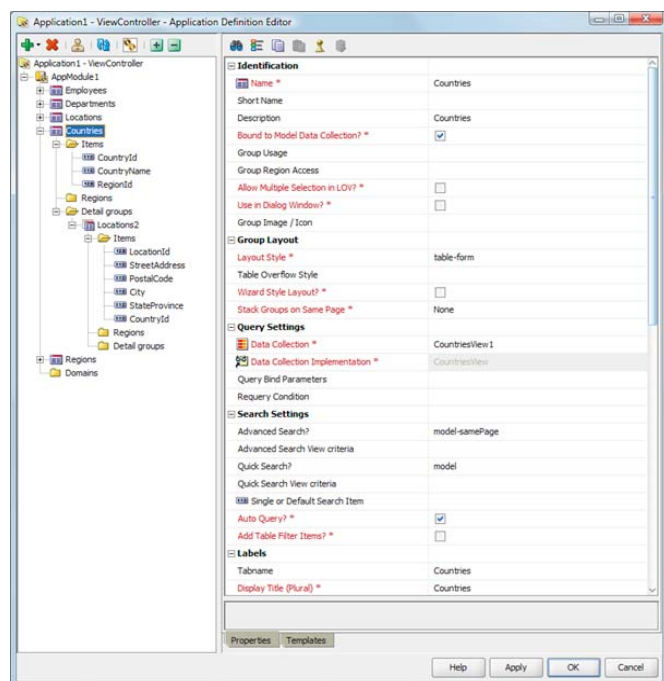
Door in het ViewController project onder de rechtermuisknop te kiezen voor 'Edit JHeadstart Application Definition' wordt de Application Definition Editor gestart (zie afbeelding 1).

JHeadstart gebruikt een Application Definition en (meerdere) Service Definitions om de structuur van de applicatie te definiëren. Deze definities worden vastgelegd in aparte xml bestanden. Zij identificeren welke pagina's nodig zijn, hoe deze pagina's gerelateerd zijn aan elkaar, hun layoutstijlen, op welke informatiebron zij gebaseerd zijn, enzovoort.

De Application Definition definieert de globale instellingen voor de gehele applicatie. Elke Service in de Application Definition zal gegenereerd worden met haar eigen menu structuur (bijv. tabbladen). Elke (top level) groep in de service correspondeert met één item in dit menu (bijv. een tabblad).

In de linkerzijde van de Application Definition Editor zijn de volgende onderdelen te onderkennen:

- **Application.** De application definition (de bovenste node in de Application Definition Editor) wordt gebruikt voor globale instellingen die de hele applicatie beïnvloeden.
- **Service.** Een service moet gezien worden als een belangrijke subset van de applicatie. Het bevat een set van logisch aan elkaar gerelateerde functionaliteit waarmee een gebruiker activiteiten uitvoert die logisch met elkaar verbonden zijn.



Afbeelding 1: Application Definition Editor.

- **Groups.** Een service bestaat uit één of meerdere groepen. Een groep maakt het gebruikers mogelijk om een enkele data collectie, die gekoppeld is aan View Object, uit te vragen en aan te passen.
- **Items.** Een item is een koppeling naar een attribuut van een View Object, welke meestal gekoppeld is aan een kolom van een tabel.
- **Lists of Values.** Een List of Values (LOV) constructie in een Application Definition koppeld een item aan een groep van het type LOV, en specificeert welke items in de LOV groep gekoppeld moeten worden aan welke items in de basis groep.
- **Regions.** De regions node kan drie verschillende object typen bevatten:
 - Item Regions. Een item region maakt het mogelijk om items te groeperen in een benoemd gedeelte (region) van een pagina.
 - Detail Group Regions. Een detail group region kan gebruikt worden om een specifieke region aan te maken voor een detail group (geneste groep). Op deze manier is er meer controle mogelijk over de plaatsing van de detail groep op de pagina.
 - Region Containers. Een region container is, zoals de naam zegt, een container voor regions. De Regions node is een voorbeeld van een region container. Hier kan een Layout Style (horizontal, vertical en stacked) worden ingesteld.
- **Detail Groups.** Groepen kunnen genest worden om master-detail (parent-child) relaties aan te maken.
- **Domains.** Een domain is een (short) list van waarden die normaliter gebruikt worden om een dropdown list te vullen. Er worden twee soorten onderscheiden: static en dynamic.

Ability to Generate Page Fragments and (Dynamic) Regions

Met behulp van de nieuwe eigenschappen 'Default Group Usage' en 'Default Group Region Access' kan er gespecificeerd worden hoe groepen (groups) worden gegenereerd. Net als in versie 10.1.3 kan men groepen genereren als stand-alone pagina's (Default Group Usage = Stand-alone Pages). Om echter de herbruikbaarheid van de groepen te vergroten, wordt er aanbevolen om ze te genereren als paginafragmenten binnen een region (Default Group Usage = Region with Page Fragments). Daarbij heb je de mogelijkheid om één common UI shell pagina te genereren ('One-page application'), waarbij alle group regions getoond worden door middel van een dynamic region (Default Group Region Access = Common UI Shell Page), of om aparte groep-specifieke UI Shell pagina's te genereren (Default Group Region Access = Group UI Shell Page). Bij de voorbeelden in dit artikel is gekozen voor dit laatste.

Generation of (un)Bounded Task Flows

JHeadstart gebruikt nu de nieuwe ADF 11 controller faciliteiten

die een uitbreiding zijn van de JSF controller functionaliteit. Voor elke top-level groep wordt een bounded task flow gegenereerd. Met behulp van het nieuwe metadata element Group Parameters kunnen er specifieke custom taskflow parameters gespecificeerd worden. De gegenereerde task flows bevatten standaard al een aantal parameters om hergebruik van de task-flow te maximaliseren. Zo kunnen er bijvoorbeeld parameters worden ingesteld om een query uit te voeren of om een specifieke rij als de current row in te stellen of om de flow op te starten in 'create mode' (zinnig in wizard-style layouts). De JHeadstart Application Generator (JAG) gebruikt Velocity, een open source op Java gebaseerde template engine van de Apache Foundation. Door in het ViewController project onder de rechtermuisknop te kiezen voor 'Run JHeadstart Application Generator' wordt de applicatie gegenereerd. De top-level XML source files van een application staan bekend als application's bootstrap configuration files. De belangrijkste bootstrap XML source file heeft meestal als naam adfc-config.xml. Wanneer er een nieuwe ADF task flow wordt aangemaakt, zal JDeveloper deze automatisch als verwijzing toevoegen in adfc-config.xml zodat deze onderdeel uitmaakt van de bootstrap list. Hieronder is de code te zien van de adfc-config.xml:

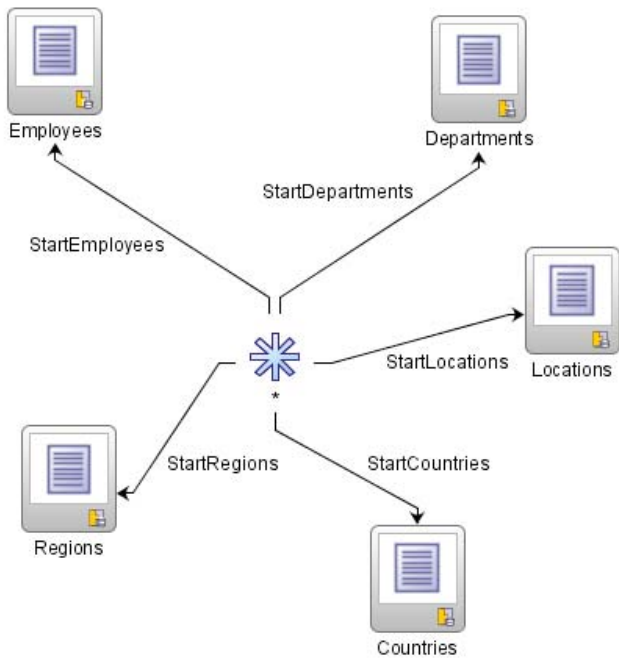
```
<?xml version="1.0" encoding="windows-1252" ?>
<adfc-config xmlns="http://xmlns.oracle.com/adf/controller"
version="1.2">
  <metadata-resource>/WEB-INF/adfc-config-appmodule1.xml</metadata-
resource>
  <metadata-resource>/WEB-INF/jhs-bounded-taskflow-template.xml</metada
ta-resource>
  <metadata-resource>/WEB-INF/jhs-fragment-taskflow-template.xml</meta
data-resource>
  <metadata-resource>/WEB-INF/jhs-lov-taskflow-template.xml</metadata
resource>
  <metadata-resource>/WEB-INF/JhsCommon-beans.xml</metadata-resource>
  <metadata-resource>/WEB-INF/AppModule1-Breadcrumb-beans.xml</metadata-
resource>
  <metadata-resource>/WEB-INF/adfc-config-Employees.xml</metadata-
resource>
  <metadata-resource>/WEB-INF/adfc-config-Departments.xml</metadata-
resource>

  <metadata-resource>/WEB-INF/adfc-config-Locations.xml</metadata-
resource>
  <metadata-resource>/WEB-INF/adfc-config-Countries.xml</metadata-
resource>
  <metadata-resource>/WEB-INF/adfc-config-Regions.xml</metadata-
resource>
</adfc-config>
```

Unbound task flow

De unbounded task flow (zie afbeelding 2) wordt gegenereerd op basis van template ..\ViewController\templates\default\misc\facesConfig\serviceAdfcConfig.vm.

Indien er gegenereerd wordt als stand-alone pagina's, zijn er task flow calls gegenereerd, zoals bijvoorbeeld "StartCountries" (welke is gekoppeld aan bounded task flow adfc-config-Countries.xml).



Afbeelding 2: Task Flow diagram adfc-config-appmodule1.xml (pagina fragmenten binnen een region).

```
<executables>
  <taskFlow id="CountriesTaskFlow"
    taskFlowId="/WEB-INF/adfc-config-Countries.xml#CountriesTaskFlow"
    parametersMap="{pageFlowScope.CountriesTaskFlowConfig.params}"

  RefreshCondition="{pageFlowScope.CountriesTaskFlowConfig.params Changed}"

  xmlns="http://xmlns.oracle.com/adf/controller/binding"/>
</executables>
<bindings>
  <action id="Commit" RequiresUpdateModel="true"
    Action="commitTransaction"
    DataControl="AppModule1DataControl"/>
  <action id="Rollback" RequiresUpdateModel="false"
    Action="rollbackTransaction" DataControl="AppModule1DataControl"/>
</bindings>
</pageDefinition>
```

Bounded task flow

De bounded task flow (zie afbeelding 3) wordt gegenereerd op basis van template ..\ViewController\templates\default\misc\facesConfiggroupAdfcConfig.vm.

```
<task-flow-call id="StartCountries">
  <task-flow-reference>
    <document>/WEB-INF/adfc-config-Countries.xml</document>
    <id>CountriesTaskFlow</id>
  </task-flow-reference>
</task-flow-call>
```

Indien er gegenereerd wordt als pagina fragmenten binnen een region, zijn er Views gegenereerd, zoals bijvoorbeeld 'Countries' (welke is gekoppeld aan /pages/Countries.jspx).

```
<view id="Countries">
  <page>/pages/Countries.jspx</page>
</view>
```

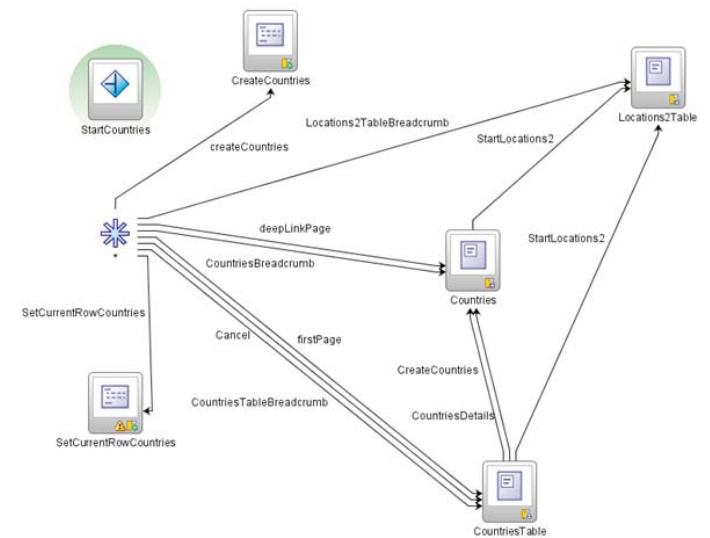
In deze pagina wordt een bounded task flow als ADF region gebruikt.

```
<f:facet name="pageContent">
  <af:region id="mainRegion" value="#{bindings.CountriesTaskFlow.region Model}" partialTriggers="::Menu1 ::Menu2 ::Dmenu0 ::Dmenu1 ::Dmenu2 ::pendingChangesDialog" />
</f:facet>
```

In de bijbehorende page definition (CountriesUIShellPageDef.xml) is de task flow binding voor de region aangemaakt.

```
<?xml version="1.0" encoding="UTF-8" ?>
<pageDefinition xmlns="http://xmlns.oracle.com/adfm/uiModel"
  version="11.1.1.54.7" id="CountriesUIShellPageDef"
  Package="application1.view.pageDefs"

  ControllerClass="oracle.jheadstart.controller.jsf.lifecycle.JhsPageLifecycle"
  SkipValidation="true">
  <parameters/>
```



Afbeelding 3: Task Flow diagram adfc-config-Countries.xml (pagina fragmenten binnen een region).

Hierbij worden de volgende activiteiten gegenereerd:

- Router 'StartCountries' (default activity).
- Method Call 'CreateCountries'.
- Method Call 'SetCurrentRowCountries'.
- View 'CountriesTable'.
- View 'Countries'.
- View 'Locations2Table'.

Indien er gegenereerd wordt als stand-alone pagina's, worden de volgende view definities gegenereerd:

```
<view id="CountriesTable">
  <page>/pages/CountriesTable.jsp</page>
</view>
<view id="Countries">
  <page>/pages/Countries.jsp</page>
</view>
<view id="Locations2Table">
  <page>/pages/Locations2Table.jsp</page>
</view>
```

Indien er gegenereerd wordt als pagina fragmenten binnen een region, worden de volgende view definities gegenereerd:

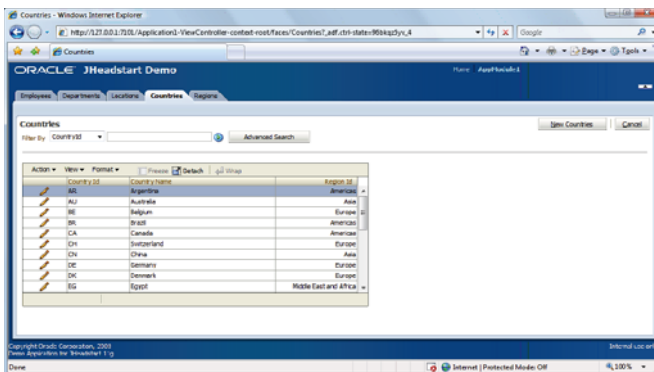
```
<view id="CountriesTable">
  <page>/pages/CountriesTable.jsff</page>
</view>
<view id="Countries">
  <page>/pages/Countries.jsff</page>
</view>
<view id="Locations2Table">
  <page>/pages/Locations2Table.jsff</page>
</view>
```

Default look and feel

Indien de applicatie wordt gegenereerd met de default look and feel, dan is er per (top level) groep (bijvoorbeeld Countries) in de service (AppModule1) een tabblad gegenereerd. Een klik op het tabblad (menu item) zal een actie starten in de unbounded task flow, zoals bijvoorbeeld 'StartCountries'.

Indien er gegenereerd is als stand-alone pagina's, zal bij de Countries pagina als URL zichtbaar zijn: <http://127.0.0.1:7101/Application1-ViewController-context-root/faces/CountriesTaskFlow/CountriesTable>?

Indien er gegenereerd is als pagina fragmenten binnen een region, zal bij de Countries pagina als URL zichtbaar zijn: <http://127.0.0.1:7101/Application1-ViewController-context-root/faces/Countries>. In afbeelding 4 is de Countries pagina te zien (door middel van pagina fragment CountriesTable.jsff), met een tabel met countries.



Afbeelding 4: Pagina Countries (paginafragmenten binnen een region).

In de Countries pagina, is een knop 'New Countries' te zien, die als actie 'CreateCountries' heeft (zie ook de bounded task flow). Na een klik op deze knop verschijnt de Enter New Countries pagina (door middel van pagina fragment Countries.jsff).

Indien in de tabel met countries van de Countries pagina, de details opgevraagd worden van een bepaalde country, dan zal de actie 'CountriesDetails' gestart worden. De details van de eerder geselecteerde country zijn te zien door middel van pagina fragment Countries.jsff. In deze pagina (Edit Countries) is de knop 'Locations' te zien, die als actie 'StartLocations2' heeft, waarmee een tabel met locaties getoond kunnen worden.

Pagina fragment Countries.jsff wordt dus zowel voor het aanmaken als het wijzigen van countries gebruikt. Afhankelijk van pageFlowScope variabelen worden bepaalde componenten van de pagina anders ingesteld. Zo wordt bijvoorbeeld de Locations knop niet getoond (rendered) in de Enter New Countries pagina.

```
<af:commandButton textAndAccessKey="#{nls['COUNTRIES_TO_LOCATIONS2_BUTTON']}"
  action="StartLocations2"
  rendered="#{!pageFlowScope.createModes.CreateCountries and (bindings.CountriesIterator.currentRowIndexInRange!=-1 and bindings.CountriesIterator.findMode!='true')}" />
```

Net zoals in de Edit Countries pagina via een knop een tabel met locaties getoond kunnen worden, kunnen in de Edit Regions pagina via een knop een tabel met countries getoond worden behorend bij bijvoorbeeld Europa (door middel van paginafragment Countries2Table.jsff).

Samenvatting

In dit artikel is nader ingegaan op enkele nieuwe functionaliteiten van Oracle JHeadstart 11g in combinatie met ADF 11g. Naast het gebruik van het ADF controller framework (met unbounded en bounded task flows) is ook het gebruik van pagina fragmenten binnen een region nader toegelicht. Gesteld kan worden dat de nieuwe release van Oracle JHeadstart goed gebruik maakt van de functionaliteiten die ADF 11g biedt.

Links en referenties

- <http://www.oracle.com/technology/products/jheadstart/index.html>
- http://blogs.oracle.com/jheadstart/2009/03/jheadstart_1111_technical_prev.html
- http://blogs.oracle.com/jheadstart/2009/08/oracle_jheadstart_1111_techinc.html
- JHeadstart 11.1.1 Technical Preview, Steven Davelaar, March 6, 2009.
- JHeadstart 11.1.1 Technical Preview 2, Steven Davelaar, August 12, 2009.
- ORACLE JHEADSTART 11g for ADF (RELEASE 11.1.1 - DRAFT), DEVELOPER'S GUIDE, MARCH 2009.

Een gratis evaluatieversie van JHeadstart 11g is nog niet beschikbaar, wel is nog steeds de 10.1.3 evaluatieversie gratis te downloaden.



Marc Lameriks is Oracle CoP leader en thought leader van het Oracle ADF Competence Center bij Capgemini en op projecten werkzaam als Software Architect (e-mail: marc.lameriks@capgemini.com).