

Oracle JDeveloper en ADF 11g

Een sprong voorwaarts

Met de komst van Oracle JDeveloper en ADF 11g is volgens Oracle een grote sprong voorwaarts gemaakt in de snelheid waarmee ontwikkelaars componenten kunnen aanmaken, samenstellen en hergebruiken om zodoende rijke, interactieve applicaties te bouwen die een hoge performance hebben en makkelijk te onderhouden zijn. In dit artikel zal op enkele nieuwe functionaliteiten, met name het ADF controller framework, nader worden ingegaan.

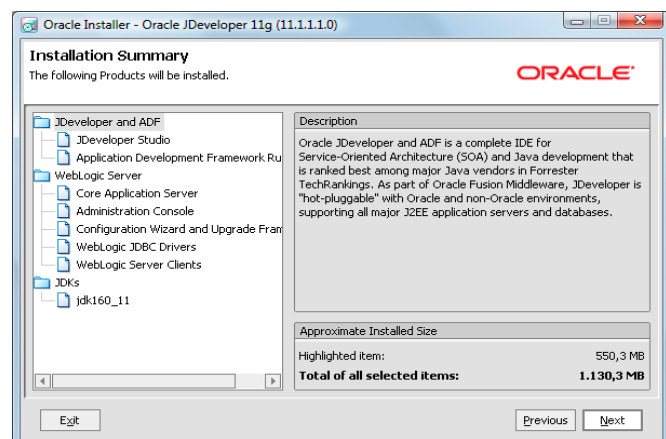
Oracle JDeveloper 11g is een complete geïntegreerde ontwikkelomgeving (IDE) voor Service-Oriented Architecture (SOA), Java en de ontwikkeling van Rich Enterprise Application's (REA). Als onderdeel van Oracle Fusion Middleware 11g, is JDeveloper 11g 'hot-pluggable' met Oracle en niet-Oracle omgevingen, en biedt het ondersteuning aan alle belangrijke J2EE applicatie servers en databases.

Oracle ADF is een end-to-end ontwikkel framework, gebouwd boven op het J2EE platform en biedt ongeëvenaarde productiviteit aan applicatie ontwikkelaars. Het framework levert geïntegreerde infrastructuur oplossingen voor de diverse lagen van een applicatie en een eenvoudige manier om daar bovenop te ontwikkelen.

Oracle JDeveloper 11g brengt nieuwe functionaliteit om een volledige ontwikkelomgeving voor Java EE 5, inclusief EJB 3.0 en JSF 1.2, te leveren. Tevens bevat het nieuwe functionaliteit voor AJAX ontwikkelaars, zoals een nieuwe set van ADF Faces rich client componenten alsook Javascript editing en debugging. De ontwikkeling van web services is sterk verbeterd met ondersteuning voor JAX-WS, een compleet webservice test framework en een veelomvattende WSDL editor. Er zijn voor ontwikkelaars ook vele productiviteitsverbeteringen in de IDE en Java ontwikkel toepassingen te ontdekken. Het bouwen van applicaties met Oracle ADF is drastisch verbeterd met de toevoeging van ADF Faces rich client, een nieuw JSF-gebaseerd ADF controller framework en vele productiviteit verhogende verbeteringen.

De belangrijkste nieuwe functionaliteiten zijn:

- JDeveloper Core IDE.
- Java Coding and Agile Development.
- Debugging and Profiling.
- Database Development.
- Java EE Application Server Integration and Deployment.
- UML Development.
- Web Services Development.
- Web and Ajax Development.
- XML Development.
- EJB 3.0 Development.
- Oracle TopLink Development.
- ADF Framework.
- ADF Faces Rich Client.
- ADF Databinding.
- ADF Business Components.
- ADF Controller (Task Flow).
- ADF Swing.
- ADF Data Visualization.
- Customization and Personalization.
- ADF Desktop Integration.
- ADF Mobile.
- Oracle Team Productivity Center.

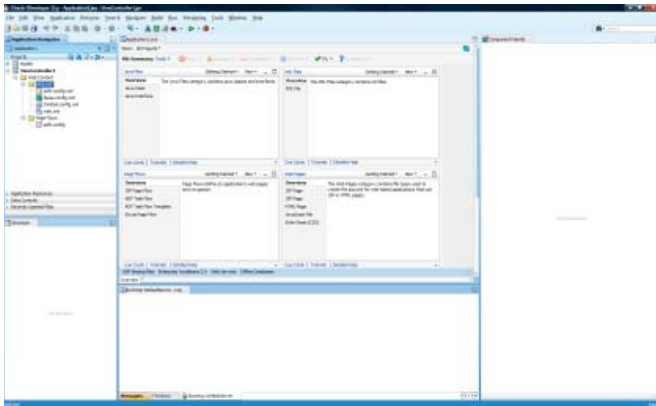


Afbeelding 1: Oracle Installer

JDeveloper Core IDE

JDeveloper 11g maakt gebruik van een installer om fouten gedurende de installatie te voorkomen, waarbij de installatie in een nieuwe of een bestaande Middleware home wordt uitgevoerd (zie afbeelding 1).

In de hele IDE zijn verbeteringen aangebracht om het werken met JDeveloper nog makkelijker te maken (zie afbeelding 2).



Afbeelding 2: Oracle JDeveloper 11g IDE.

Server Integration

JDeveloper 11g bevat, zoals in vorige versies, een geïntegreerde applicatie server om zodoende applicaties die een middle tier server nodig hebben, te kunnen runnen en debuggen. In deze release wordt Oracle WebLogic Server (WLS) als geïntegreerde applicatie server gebruikt. Bij de installatie van JDeveloper wordt een volledige WLS server en een domain 'DefaultDomain' geïnstalleerd, welke geconfigureerd is voor het draaien van ADF applicaties. Bij het runnen of debuggen van een applicatie zal de IDE, in plaats van de hele server te stoppen en starten, eenvoudigweg de applicatie deployen naar de geïntegreerde applicatie server. Hiermee wordt voor gebruikers de wachttijd vermindert gedurende het ontwikkelproces.

ADF Task Flows

ADF task flows leveren een modulaire aanpak voor het definiëren van de control flow in een applicatie. In plaats van een applicatie te representeren als een enkele grote JSF page flow, kan deze opgedeeld worden in verzamelingen van herbruikbare task flows. In elke task flow worden applicatie activiteiten gedefinieerd, die elk een bepaald stukje werk verrichten, alsmede control flows tussen activiteiten.

Een ADF task flow biedt de volgende voordelen bovenop een standaard JSF page flow:

- De applicatie kan opgedeeld worden in een serie van taken, in plaats van de applicatie te representeren als een enkele JSF page flow.
- Activiteiten, anders dan pagina's kunnen worden gebruikt in een task flow.

- Er kan worden genavigeerd tussen activiteiten, niet alleen tussen pagina's.
- Task flows zijn modulair en herbruikbaar binnen dezelfde applicatie of een andere applicatie.
- Shared memory scope maakt het mogelijk data uit te wisselen tussen activiteiten binnen de task flow.

De ADF controller ondersteunt twee typen task flows:

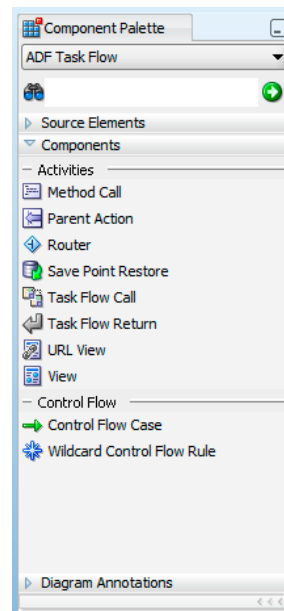
- Unbounded task flows.
- Bounded task flows.

Een task flow bestaat uit activiteiten en control flow cases, die de overgangen definiëren tussen activiteiten. Zo kan een control flow genaamd startView3 een

overgang definiëren tussen activiteit View2 en activiteit View3.

Sinds de komst van JDeveloper 11g zijn er nieuwe task flow activiteiten toegevoegd. Ook is er een nieuw control flow element genaamd Wildcard Control Flow Rule.

Hieronder volgt een korte toelichting op de activiteiten en control flow elementen van het ADF controller framework (zie afbeelding 3). Sommige activiteiten zijn alleen zichtbaar in een bounded task flow.



Afbeelding 3: Component Palette, ADF Task Flow.

Method Call

Met een method call activiteit kan er, vanuit elke willekeurige plek binnen een applicatie control flow, een zelfgemaakte of een standaard ingebouwde method aangeroepen worden, die applicatie logica uitvoert. Zo kunnen bijvoorbeeld methods gespecificeerd worden om taken uit te voeren zoals initialisatie voorafgaand aan het tonen van een pagina, opschoning na het afsluiten van een pagina, foutafhandeling, enz.

Parent Action

Vanuit een bounded task flow, welke onderdeel uitmaakt van een ADF region, zou het nodig kunnen zijn om navigatie te activeren in de omhullende view. Een parent action activiteit maakt het een ADF bounded task flow mogelijk om outcomes te genereren die doorgegeven worden naar zijn parent. Deze outcomes worden gebruikt om te navigeren binnen de task flow die de omhullende view bevat, in plaats van te navigeren binnen de task flow van de ADF region.

Router

Een router activiteit kan gebruikt worden om declaratief de control te routeren naar bepaalde activiteiten, gebaseerd op logica welke gespecificeerd is in een Expression Language (EL) expressie.

Save Point Restore

Met een save point restore activiteit is het mogelijk om terug te keren naar een eerder aangemaakt persistent save point in een applicatie die 'save for later' functionaliteit ondersteunt. Een save point maakt een snapshot van een Fusion web applicatie op een specifiek moment. Save point restore maakt het de applicatie mogelijk terug te keren naar alles dat was bewaard op het moment dat de save point oorspronkelijk was aangemaakt.

Task Flow Call

Een task flow call kan gebruikt worden om een bounded task flow aan te roepen, zowel vanuit een unbounded als een bounded task flow. Bepaalde opties op te geven bij de task flow call activiteit maken het mogelijk een bounded task flow aan te roepen binnen dezelfde applicatie of een andere applicatie.

Task Flow Return

Een task flow return activiteit identificeert het punt in de applicatie control flow waar een bounded task flow eindigt en stuurt de control flow terug naar de aanroepende activiteit. Een task flow return activiteit kan alleen gebruikt worden in een bounded task flow. Een grijze cirkel rond een activiteit icon geeft aan dat de activiteit een exit point is voor een bounded task flow. Elke bounded task flow kan nul tot meerdere task flow return activiteiten bevatten.

URL View

Een URL view activiteit kan gebruikt worden om een redirect uit te voeren van de root view port (bijvoorbeeld een browser pagina) naar elke URL-adresseerbare resource, zelfs vanuit de context van een ADF region. URL-adresseerbare resource zijn onder andere: ADF bounded task flows; view activiteiten in een ADF unbounded task flow en; adressen extern ten opzichte van de huidige web applicatie (bijvoorbeeld Google)

View

Het belangrijkste type task flow activiteit is een view, die een JSF pagina of paginafragment toont. Een paginafragment is een JSF JSP document dat getoond wordt als content in een andere JSF pagina. Paginafragmenten worden doorgaans gebruikt in bounded task flows. De bounded task flow kan aan een pagina worden toegevoegd als ADF region.

Control Flow Case

In een control flow case wordt vastgelegd hoe de control overgaat van de ene activiteit naar de volgende in de applicatie.

Wanneer een control flow case tussen twee activiteiten wordt aangemaakt, ontstaat automatisch een control flow bestaande uit het volgende:

- Control-flow-rule: Identificeert de source activiteit door middel van een from-activity-id.
- Control-flow-case: Identificeert de target activiteit door middel van een to-activity-id.

Wanneer een activiteit als de source voor een control flow case is aangemerkt, worden alle additionele control flow case die dezelfde activiteit als source hebben onder dezelfde control flow rule geplaatst.

```
<control-flow-rule>
  <from-activity-id>View2</from-activity-id>
  <control-flow-case>
    <from-outcome>startView3</from-outcome>
    <to-activity-id>View3</to-activity-id>
  </control-flow-case>
</control-flow-rule>
```

Wildcard Control Flow Rule

Een wildcard control flow rule representeert een control flow from-activity-id die een trailing wildcard (foo*) bevat of een enkelvoudig wildcard teken (*). In het laatste geval wordt daarmee aangeduid dat de control kan overgaan naar de activiteiten welke er aan verbonden zijn in het task flow diagram, vanaf elke willekeurige activiteit in de task flow.

Aan de hand van enkele voorbeelden zal het gebruik van het ADF controller framework nader worden toegelicht.

Unbounded task flow

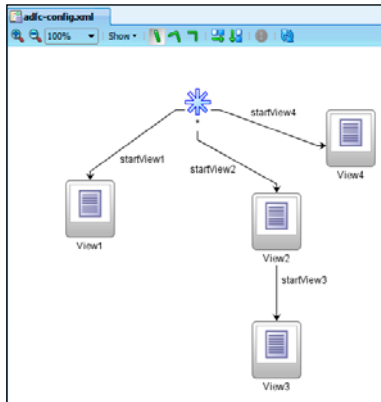
Alhoewel er meerdere bounded task flows in een applicatie aanwezig kunnen zijn, kan er maar één unbounded task flow zijn. De unbounded task flow representeert de buitenste task flow van een ADF applicatie. Het beginpunt (point of entry) van een applicatie kan een view/pagina zijn binnen de unbounded task flow.

De kenmerken van een unbounded task flow zijn:

- Eerste toegang tot de task flow stack, de buitenste task flow.
- Geen duidelijke definieerbare begrenzing of een enkelvoudig toegangspunt tot de flow.
- Kan gebruikt worden om de "top level" flow van een applicatie te definiëren.
- Bookmarkable pagina's.

Als begin is er gekozen voor het aanmaken van een Fusion Web Application (ADF), door te kiezen voor File | New | General | Applications | Fusion Web Application (ADF). Er wordt dan door de bijbehorende wizard een Model en ViewController project aangemaakt. Binnen het ViewController project wordt standaard een unbounded task flow (adfc-config.xml) aangemaakt, welke zowel onder node WEB-INF als onder node Page Flows kan worden geopend (zie afbeelding 2).

Als voorbeeld worden aan deze unbounded task flow eerst een aantal View activiteiten toegevoegd. Ga naar het Component Palette, selecteer in de drop down list ADF Task Flow, klik op Components categorie en kies daarna onder Activiteiten voor View. Daarna zijn de control flow elementen toegevoegd. Bij iedere View activiteit wordt een uitroepteken in het bijbehorende icon getoond (met de tekst Required child missing from element view), waarmee wordt aangegeven dat er nog geen bijbehorende pagina aanwezig is. Een pagina (*.jspx) kan eenvoudig



worden aangemaakt door bij een View activiteit via de rechter muisknop te kiezen voor Create Page. Op deze manier zijn in de unbounded task flow een viertal View activiteiten aangemaakt (zie afbeelding 4).

Afbeelding 4: Task Flow diagram adfc-config.xml.

Bounded task flow

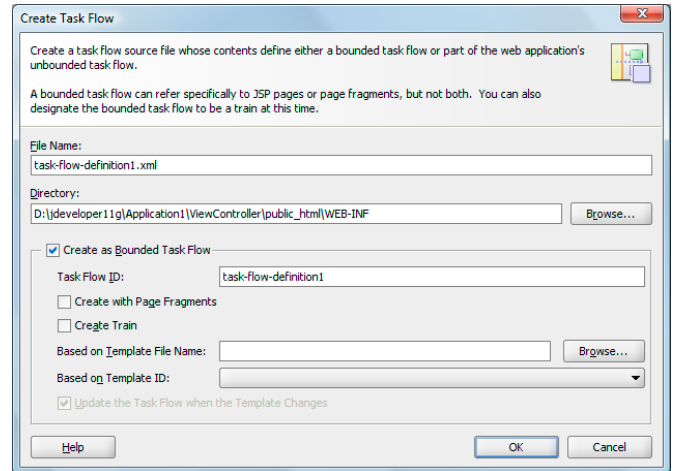
Alhoewel er maar één unbounded task flow in een applicatie aanwezig kan zijn, kunnen er meerdere bounded task flows zijn. Een bounded task flow kan gezien worden als een proces omdat het een modulaire blok van task flow functionaliteiten definieert die hergebruikt kunnen worden. Het is het enige type task flow dat als een 'region' van een pagina gebruikt kan worden.

Bounded task flows zijn modulaire blokken van task flow functionaliteiten die hergebruikt kunnen worden, met de volgende kenmerken:

- Eén enkel toegangspunt tot de flow.
- Duidelijk definieerbare begrenzing.
- Pageflow memory scope.
- Declaratief transactie management.
- Declaratieve back button support.
- Mogelijkheid om input parameters te ontvangen en waarden te retourneren.
- On demand metadata loading.
- Fragment hergebruik door task flow templates.
- Adresseerbaar/aanroepbaar.

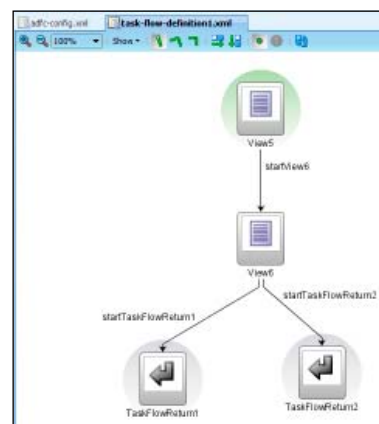
Aanroep vanuit unbounded task flow

In onderstaand voorbeeld wordt een bounded task flow aangemaakt die vanuit de unbounded task flow aangeroepen zal gaan worden. Via File | New | Web Tier | JSF | ADF Task Flow is een task flow met naam task-flow-definition1.xml aangemaakt (zie afbeelding 5).



Afbeelding 5: Create Task Flow wizard.

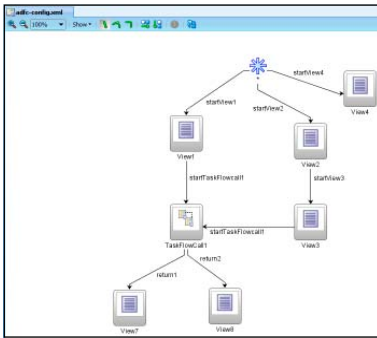
Net zoals bij de unbounded task flow zijn er een aantal View activiteiten, inclusief pagina's (.jspx), toegevoegd. Vervolgens zijn een tweetal Task Flow Return activiteiten toegevoegd en vervolgens de control flow elementen. De eerste activiteit die aan een bounded task flow wordt toegevoegd wordt automatisch als de zogenaamde default activity (met een groene cirkel rond de activiteit icon) gekenmerkt. Het is altijd mogelijk een bepaalde activiteit als de default activity in te stellen door een activiteit te selecteren en via de rechter muisknop te kiezen voor Mark Activity | Default Activity. Wanneer een bounded task flow wordt gestart is de default activity altijd de eerste activiteit die gestart wordt. Bij Task Flow Return activiteiten TaskFlowReturn1 en TaskFlowReturn2 is een bepaalde outcome ingesteld (via het



Property Inspector scherm), respectievelijk "return1" en "return2". Een Task Flow Return activiteit wordt met een grijze cirkel rond de activiteit icon weergegeven in het task flow diagram (zie afbeelding 6).

Afbeelding 6: Task flow diagram task-flow-definition1.xml.

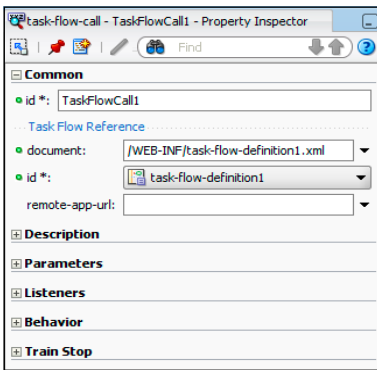
De volgende stap is om de unbounded task flow uit te breiden met de aanroep naar de bounded task flow. In de unbounded task flow is een Task Flow Call activiteit TaskFlowCall1 toegevoegd welke vanuit activiteiten View1 en View3 aan te roepen is. Afhankelijk van de outcome van de Task Flow Call activiteit zal er naar activiteit View7 of View8 worden gegaan, welke aan de unbounded task flow, inclusief pagina's (.jspx), zijn toegevoegd (zie afbeelding 7).



Afbeelding 7: Task flow diagram adfc-config.xml.

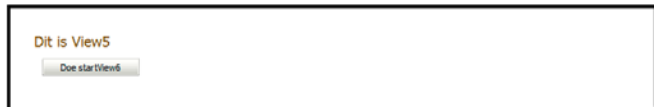
Bij de Task Flow Call activiteit TaskFlowCall1 is in het Property Inspector scherm bij onderdeel Task Flow Reference als document “/WEB-INF/task-flow-definition1.xml” opgege-

ven en als id 'task-flow-definition1' (zie afbeelding 8). Eventueel zou ook remote-app-url kunnen worden meegegeven als de te starten bounded task flow in een andere applicatie dan de huidige aanwezig is. Ook eventuele parameters, gedefinieerd in de bounded task flow, kunnen worden meegegeven.



Afbeelding 8: Property Inspector, TaskFlowCall1.

Indien Task Flow Call activiteit TaskFlowCall1 wordt gestart (bijvoorbeeld vanuit View1.jspx), wordt de default activity View5 gestart (zie afbeelding 9).



Afbeelding 9: Pagina View5.jspx.

Via de knop 'Doe startView6' kan er naar pagina View6.jspx genavigeerd worden. Vanuit deze pagina kan via de knoppen 'Doe startTaskFlowReturn1' en 'Doe startTaskFlowReturn2' naar respectievelijk pagina's View7.jspx en View8.jspx (van de unbounded task flow) genavigeerd worden (zie afbeelding 10).

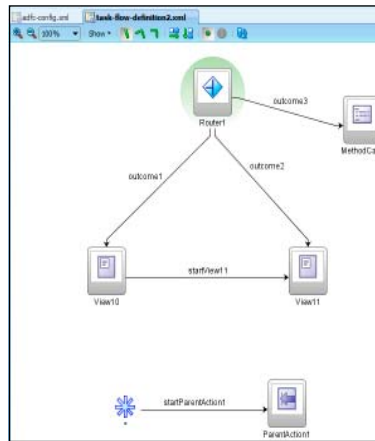


Afbeelding 10: Pagina View6.jspx.

Bounded task flow als ADF region

In onderstaand voorbeeld wordt een bounded task flow aangemaakt die als ADF region gebruikt zal gaan worden. Via File | New | Web Tier | JSF | ADF Task Flow is een task flow met naam

task-flow-definition2.xml aangemaakt. Hierbij is de optie “Create with Page Fragments” aangevinkt, zodat alle pagina's binnen de task flow als pagina fragment worden aangemaakt (zie afbeelding 11). Eventueel kan bij het aanmaken van een bounded task flow,

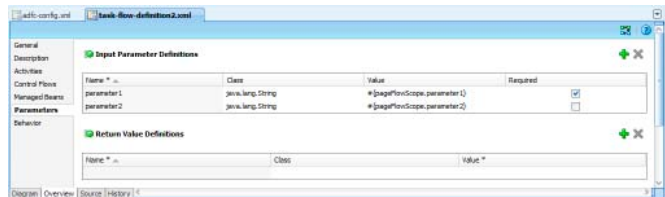


de optie “Create Train” aangevinkt worden. Een train representeert een reeks van gerelateerde activiteiten, welke een gebruiker stap voor stap begeleidt om een taak tot een succesvol einde te brengen (zie JDeveloper Help).

Afbeelding 11: Task flow diagram task-flow-definition2.xml.

In deze bounded task flow is een Router activiteit Router1 aangemaakt (als default activity), welke een drietal mogelijke outcomes kan hebben. De outcome van activiteit Router1 in dit voorbeeld is afhankelijk van bepaalde parameter waarden die meegegeven worden bij de aanroep van deze bounded task flow. De zogenaamde default outcome van activiteit Router1 is “outcome3”, welke de Method Call activiteit MethodCall1 start. Voor de overige twee outcomes zijn er View activiteiten, inclusief pagina's (.jspx), toegevoegd. Een pagina kan eenvoudig worden aangemaakt door bij een View activiteit via de rechter muisknop te kiezen voor Create Page Fragment. Tenslotte is een Parent Action activiteit ParentAction1 toegevoegd, alsmede de control flow elementen.

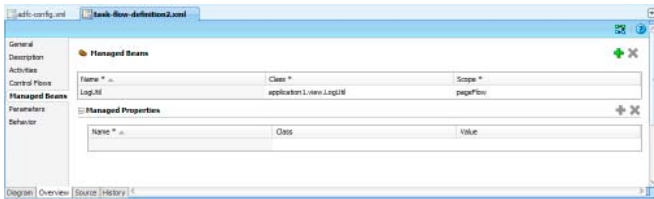
In dit voorbeeld zijn in de bounded task flow een aantal parameters gedefinieerd, om daarmee de activiteit Router1 van functionaliteit te kunnen voorzien. De parameters genaamd parameter1 en parameter2 zijn in het task flow Overview scherm onder Parameters aangemaakt. Deze parameters worden standaard in de pageFlowScope vastgelegd (zie afbeelding 12).



Afbeelding 12: Parameters in task-flow-definition2.xml.

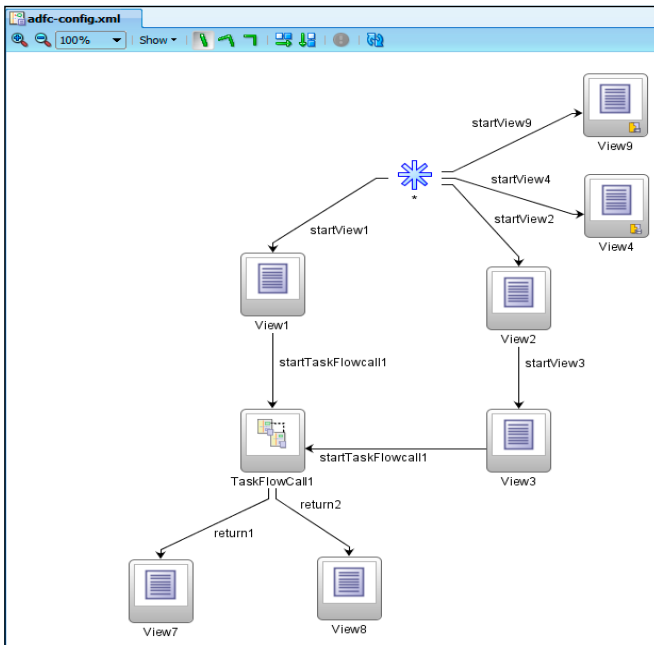
Activiteit Router1 zal in dit voorbeeld als parameter1 de waarde 'A' heeft activiteitView10 starten, bij waarde 'B' activiteit View11 starten en in overige gevallen activiteit MethodCall1 (zie task-flow-definition2.xml).

De activiteit MethodCall1 doet een aanroep naar method writeLog van class LogUtil, welke als waarde de tekst “Bounded task flow heeft een ongeldige waarde voor parameter I!” meekrijgt. Deze method zorgt er voor dat er een log melding verschijnt. De class LogUtil is hiertoe in de bounded task flow geregistreerd als Managed Bean in het task flow Overview scherm onder Managed Beans, met pageFlow als scope (zie afbeelding 13).



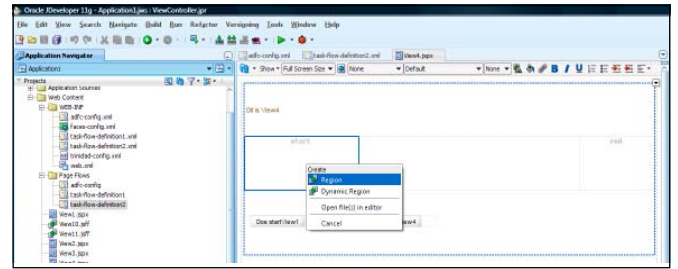
Afbeelding 13: Managed Beans in task-flow-definition2.xml.

Bij activiteit ParentAction1 is als Parent Outcome ‘startView2’ ingesteld (via het Property Inspector scherm). Met andere woorden outcome ‘startView2’ wordt getriggerd binnen de task flow die de omhullende view (van de ADF region) bevat, ofwel activiteit View2 wordt gestart in de unbounded task flow. In de volgende stappen is de unbounded task flow weer aangepast zodat de bounded task flow als ADF region gebruikt wordt. In de unbounded task flow is een View activiteit View9 toegevoegd, inclusief pagina (.jspx) en control flow elementen (zie afbeelding 14).



Afbeelding 14: Task flow diagram adfc-config.xml.

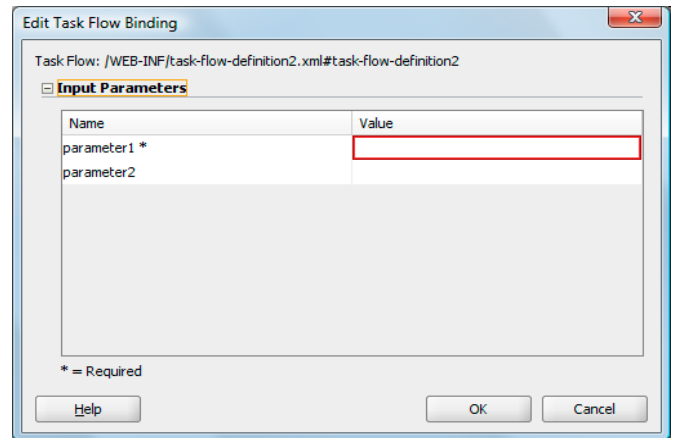
In pagina View4.jsx en View9.jsx is bounded task flow task-flow-definition2 als region toegevoegd aan de pagina, door de



Afbeelding 15: Design pagina View4.jsx.

task flow te slepen naar de pagina en te kiezen voor Create Region (zie afbeelding 15). In een pagina kan ook een ADF dynamic region aangemaakt worden, welke vergelijkbaar is met een ADF region, behalve dat de task flow ID in de task flow binding dynamisch bepaald wordt op runtime. Dat wil zeggen dat op runtime pas bepaald wordt welke bounded task flow in de ADF region gebruikt wordt (zie JDeveloper Help).

Omdat in de bounded task flow parameters zijn gedefinieerd, verschijnt er een Edit Task Flow Binding scherm waar in dit geval de parameters parameter1 en parameter2 van een waarde kunnen worden voorzien (zie afbeelding 16). De waarden voor de task flow parameters (respectievelijk ‘A’ en ‘View4’) worden uiteindelijk vastgelegd in de Oracle ADF binding container behorend bij de pagina (View4PageDef.xml).

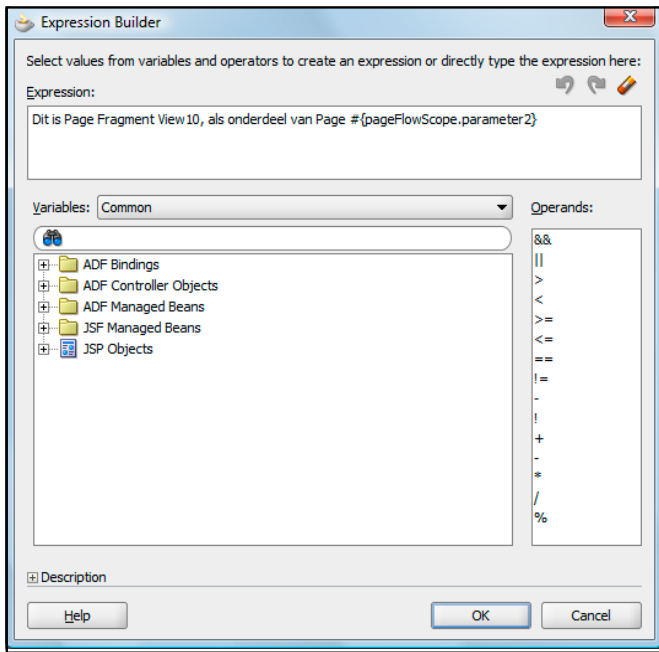


Afbeelding 16: Edit Task Flow Binding.

Indien View activiteit View4 in de unbounded task flow wordt gestart, met waarde ‘A’ voor task flow parameter I, verschijnt pagina fragment View10.jsff als ADF region binnen pagina View4.jsx (zie afbeelding 17). Het ADF region gedeelte van deze pagina is in dit voorbeeld omgeven door een stippellijn.



Afbeelding 17: Pagina View4.jsx.



Afbeelding 18: Expression Builder.

Voor het toekennen van waarden aan ADF Faces componenten in dit voorbeeld is veel gebruik gemaakt van de Expression Builder. Zo ook bijvoorbeeld voor de waarde van `af:outputText` in pagina fragment `View10.jsff` (zie afbeelding 18).

Voor `View9.jsx` is op gelijke wijze de ADF region toegevoegd, waarbij de waarden voor de task flow parameters `parameter1` en `parameter2`, respectievelijk 'B' en 'View9' zijn. De waarde van task flow parameter `parameter1` is daarbij als voorbeeld gebaseerd op de waarde van pagina parameter `parameter1` (via "`#{bindings.parameter1}`").

Indien View activiteit `View9` in de unbounded task flow wordt gestart, met waarde "B" voor task flow parameter `parameter1`, verschijnt pagina fragment `View11.jsff` als ADF region binnen pagina `View9.jsx` (zie afbeelding 19).



Afbeelding 19: Pagina View9.
jspx.

Indien View activiteit `View4` in de unbounded task flow wordt gestart, met bijvoorbeeld waarde 'Z' voor task flow parameter `parameter1`, verschijnt er geen pagina fragment als ADF region binnen pagina `View4.jsx` (zie afbeelding 20), maar de volgende log melding:

```
24-07-2009 11:03:36 DEBUG (application1.view.LogUtil) -Bounded task flow heeft een ongeldige waarde voor parameter1!
```



Afbeelding 20: Pagina View4.jsx.

Samenvatting

Naast het gebruik van een installer en Oracle WebLogic Server (WLS) als geïntegreerde applicatie server, is in dit artikel, aan de hand van enkele voorbeelden, het ADF controller framework (met unbounded en bounded task flows) nader toegelicht. Dit framework heeft de flexibiliteit en de herbruikbaarheid van componenten sterk verbeterd. Ook gezien de overige verbeteringen aan JDeveloper kan met recht gesteld worden dat Oracle JDeveloper en ADF 11g een sprong voorwaarts is. In een volgend artikel zal ik nader ingaan op het gebruik van JHeadstart in combinatie met deze nieuwe release van JDeveloper.

Nadere informatie

<http://www.oracle.com/technology/products/jdev/index.html>

Referenties

- Oracle JDeveloper 11g, Oracle data sheet.
- Oracle Application Development Framework (Oracle ADF) 11g, Oracle data sheet.
- Oracle JDeveloper and ADF Release 11g (11.1.1.0.0) New Features, October 6, 2008 and updated July 1, 2009.
- Oracle Fusion Middleware, Fusion Developer's Guide for Oracle Application Development Framework, 11g Release 1 (11.1.1), B31974-02, November 2008.

Voor de volledige code van het voorbeeld kan contact worden opgenomen met de auteur.



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).