In deze rubriek worden specifieke onderdelen van het Oracle Application Development Framework besproken. De bedoeling is dat de lezer daarna zelf met het onderwerp aan de slag kan.

ADF@WORK

Groovy in ADF-BC

Volgens het Engelse woordenboek staat Groovy voor "slang meaning cool, neat and interesting". In de ADF-wereld is het weer een taal om onder de knie te krijgen. Net als je denkt dat je met java, html, css, javascript en xml alles hebt gehad…. Daar zat je echt op te wachten. Gelukkig krijg je er wel iets voor terug.
In de praktijk blijkt het vaak lastig om afgeleide waarden, validatie en foutmeldingen snel en op een bondige manier te schrijven. Een check om te controleren of twee datums na elkaar liggen, kun je natuurlijk helemaal in Java uitschrijven.

```
((Date)getAttribute("verkoopDatum")).
compareTo((Date)getAttribute("inkoopDatum"))>0
```

Maar je zou ook kunnen overwegen om Groovy hiervoor te gebruiken.

```
verkoopDatum > inkoopDatum
```

Het resultaat is hetzelfde, maar ik weet wel waar mijn voorkeur naar uit gaat. Groovy wordt sinds release 11g van ADF ook ondersteund in ADF. De Groovy-expressies worden opgeslagen in de XML-definitie van de entity- of viewobjecten.

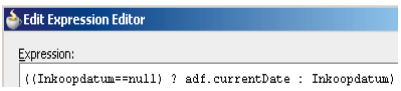
Groovy in ADF-BC

In ADF-BC kan Groovy bijvoorbeeld worden gebruikt voor het schrijven van een validatie, het samenstellen van foutmeldingen en bij het afleiden van calculated en transient attributes. Met Groovy kan dit snel en compact. Aan de hand van een eenvoudig voorbeeld op basis van een entity object 'product' wordt dit duidelijk.
Het object bevat de volgende attributen:
Aan dit object kun je met Groovy een aantal wijzigingen doorvoeren in de 'entity attribute editor'.

Code	String
Omschrijving	String
Prijsex	Number
Inkoopdatum	Date
Verkoopdatum	Date

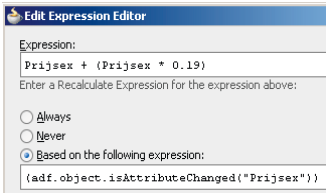
Default value: Inkoopdatum is default de datum van vandaag.
Calculated attribuut: Extra attribuut prijs inclusief BTW (19%).
Validatie: Verkoopdatum moet na de inkoopdatum liggen.
Foutmelding: Een nette foutmelding als de validatie mis gaat.

Default values



Als de Inkoopdatum leeg is, moet deze de waarde van vandaag krijgen. Met de volgende Groovy-expressie dwing je dit af.
Hier wordt gebruik gemaakt van adf.currentDate, een snelle manier om de datum van vandaag te bepalen.

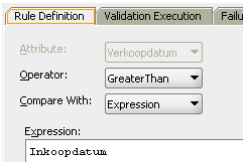
Calculated attribute



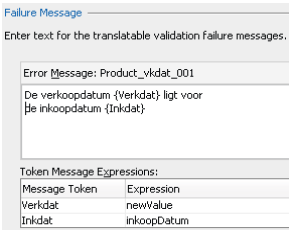
De prijs inclusief BTW kun je toevoegen door een transient attribuut op te nemen met een value-expressie waarin de prijs wordt berekend. De prijs inclusief BTW moet worden herberekend als de prijs exclusief BTW wijzigt. Ook dit kan eenvoudig worden bereikt met behulp van een Groovy expressie.

In dit geval wordt gebruik gemaakt van de 'isAttributeChanged()' eigenschap die met behulp van de Groovy expressie "adf.object.isAttributeChanged()" kan worden opgevraagd.

Validatie en foutmeldingen



Op een attribuut van een entity object kan een validatie worden opgegeven. Zoals gezegd, de verkoopdatum moet ná de inkoopdatum liggen. Een simpele check die eenvoudig te implementeren is.



Indien deze controle een fout oplevert dan kan met behulp van Groovy ook een foutmelding worden gegeven. Deze melding kan worden ingevoerd op het tabblad "failure message" van de validation rule editor.

De variabelen in de melding worden gevuld op basis van de Groovy expressies 'newValue' en 'inkoopDatum'.

Tot slot

Groovy is inderdaad 'cool, neat and interesting'. Het is een nieuwe taal, maar zeker een die het leren waard is. Je hoeft Groovy niet volledig te beheersen om een start te maken met het gebruik ervan in ADF. Start met een eenvoudig entity object, verrijk dit met behulp van Groovy-expressies en bekijk het resultaat direct in de Business Component tester.

Luc Bors (luc.bors@amis.nl) werkt als technisch specialist/architect en is ADF Expertise Lead bij AMIS Services.