

Tagebuch eines Entwicklers: Von Oracle Forms hin zu ADF

Jan-Peter Timmermann
Trivadis
Hamburg

Schlüsselworte:

Oracle Forms, Oracle ADF, Deklarative Entwicklung, JDeveloper

Einleitung

Dieser Vortrag beschäftigt sich mit der Frage, wie sich die Arbeitsumgebung für *Oracle Forms*-Entwickler verändert hat und welche Möglichkeiten es gibt, auf bisher erworbenen Kenntnissen aufzubauen, um in Zukunft weiterhin Anwendungen für Oracle entwickeln zu können. Themen, die von mir behandelt werden, sind *Oracle ADF* (Application Development Framework) und die *JDeveloper* Entwicklungsumgebung.

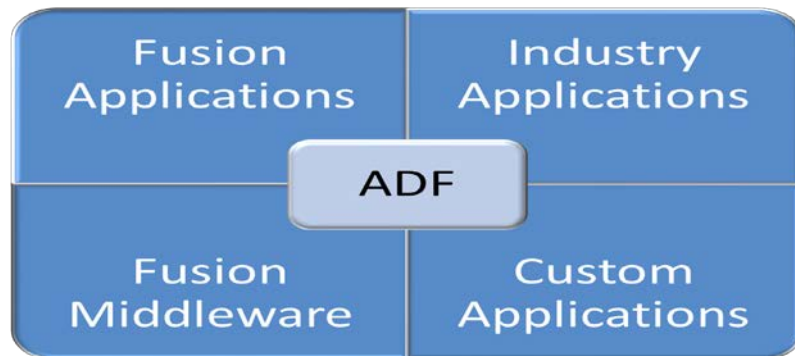
Ausgangslage

Seit über 12 Jahren beschäftige ich mich mit der Entwicklung von Anwendungen auf der Basis von *Oracle Forms* und *Reports*. In dieser Zeit habe ich mehrere Migrationen von einer Version auf die nächste miterlebt. Durch meine persönlichen Gespräche mit Kunden ist mir bekannt, dass es auf deren Seite eine große Verunsicherung hinsichtlich des zukünftigen Supports und der Weiterentwicklung von *Oracle Forms* gab. Viele Kunden haben sich deswegen von Oracle abgewandt und sind in Richtung *.NET* abgewandert. Dazu beigetragen hat auch die Neuausrichtung der Technologie weg von Client-Anwendungen hin zu Web-Anwendungen. Als es plötzlich keinen installierbaren Client mehr gab, wussten viele Kunden nicht mehr, wie sie bestehende Anwendungen auch in Zukunft weiter betreiben werden können. Außerdem wurden die neuen web-basierten Anwendungen nicht als brauchbare Alternative wahrgenommen, da die angeblichen Vorteile dieser Plattform nicht vorhanden oder zumindest nicht klar erkennbar waren. Es war abzusehen, dass neue Anwendungen auf Basis von *Oracle Forms* bald nicht mehr benötigt würden, und daher stellte sich für mich immer öfter die Frage, wie ich Zukunft weiter entwickeln kann.

Entscheidung für ADF

Ich entschied, mich auf die Suche nach neuen, zukunftssichereren Entwicklungsmöglichkeiten zu machen, vielleicht fände ich ja eine 4GL-ähnliche Entwicklungsumgebung. Immerhin wollte ich nicht jedes Jahr etwas komplett Neues lernen müssen. Ebenso wollte ich mich nicht auf ein einziges Betriebssystem beschränken müssen, sondern lieber Cross-Platform-Anwendungen entwickeln. Sehr früh erkannte ich in Oracles neuem Framework *ADF* die Möglichkeit, meine Wünsche zu erfüllen. Leider waren die ersten Ansätze von *ADF* nicht immer ausgereift. Es passierte durchaus, dass Oracle die Technik wieder komplett über den Haufen warf und keine Migrationsmöglichkeiten anbot. Größere Veränderungen machten sich erst bemerkbar, nachdem Oracle begann, seine E-Business Suite auf neue Beine zu stellen. Bis dahin war das grundlegende Framework immer noch *Forms*. Dann hatte sich Oracle entschlossen, den Nachfolger, genannt „Fusion Apps“, auf ein komplett neues Fundament zu setzen. In der Folge kam mit *ADF* ein Framework heraus, welches viele Anforderungen für die gemeinsame Entwicklung unterschiedlicher Oracle-Anwendungen erfüllt und dafür eine einheitliche Oberfläche, Infrastruktur und Tools bereitstellt.

Heute kann man mit Gewissheit sagen, dass *ADF* Oracles strategisches Framework für die Softwareentwicklung darstellt.



Komponenten, die in Fusion Apps auf Basis von *ADF* realisiert wurden, sind: Datenbank Anbindung mit *ADF Business Components*, Binding mit *ADF Model*, Web User Interface mit *ADF Faces*.

Erster Tag: Vorbereitung

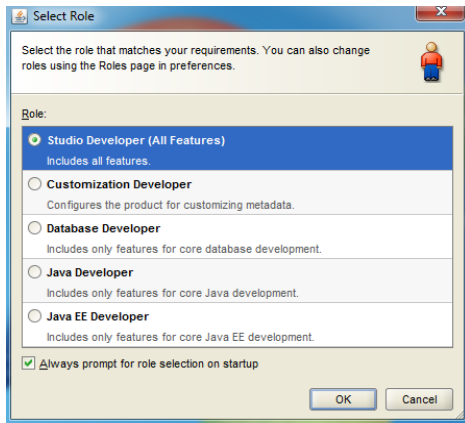
Nachdem ich meine persönliche Entscheidung getroffen hatte, begann ich mit dem Zusammentragen der benötigten Installationsmedien. Ich brauchte mit Sicherheit eine Datenbank, außerdem *JDeveloper* als Entwicklungsumgebung. Letztere ist mit 1,2 GB ziemlich groß für einen Download, aber zum Glück kommt die Version 11.1.2.0 mit allem Notwendigem daher.

Der erste Tag verging weitgehend damit, eine Datenbank (*Oracle XE* in der Version 10gR2), sowie *JDeveloper* zu installieren. Überrascht war ich dann doch, wie einfach sich das ganze gestaltet hat. Man muss nur aufpassen, dass die Installation mit ausreichenden Rechten ausgeführt wird.

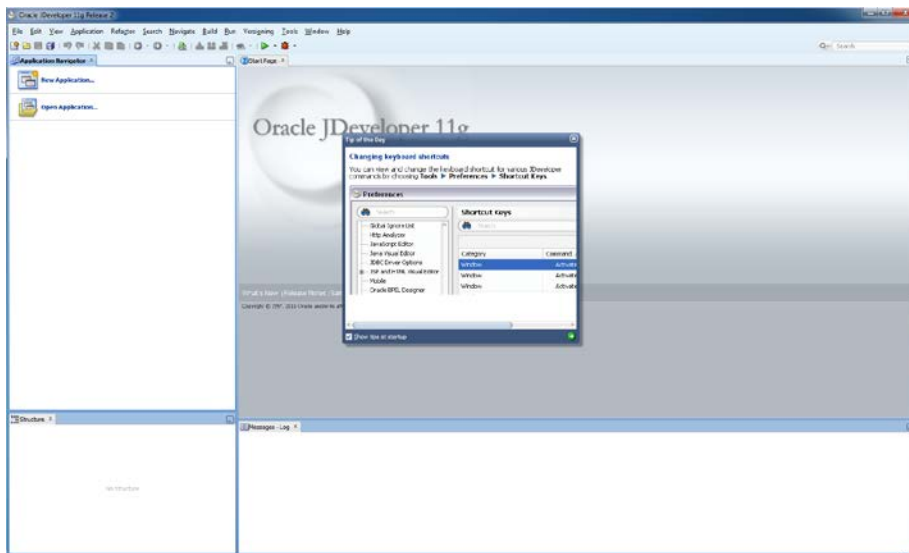
Zweiter Tag: Meine ersten Schritte mit JDeveloper

Nachdem alles installiert war, ging ich mit ziemlich großer Freude an *JDeveloper* heran. Nach einem Doppelklick auf das Desktop-Icon stellte ich mich auf eine gewisse Wartezeit ein und wollte gerade damit anfangen, meinen Kaffeebecher wieder aufzufüllen. Allerdings wurde ich überrascht, denn das Starten des *JDeveloper* 11gR2 geht recht zügig von statten.

Der erste Fensterdialog fragte mich nach der Auswahl einer Rolle:



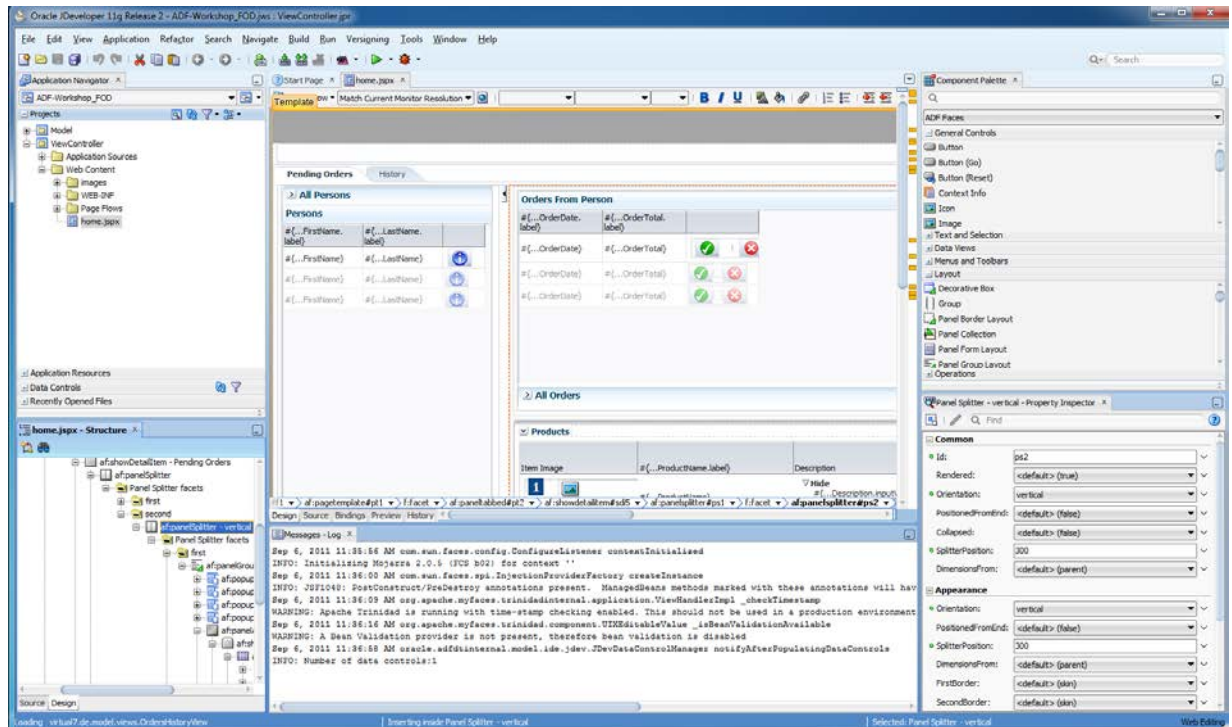
Leider wusste ich an dieser Stelle zunächst nicht, welche Rolle für mich die richtige ist. Heute kann ich sagen, dass hiermit die verfügbaren Funktionen der IDE modifiziert werden. Wenn man alle Funktionen des *JDeveloper* nutzen möchte, ist die Default-Auswahl passend. Die "Customization Developer" Rolle erlaubt genauere Anpassungsmöglichkeiten. Bei meinem ersten Versuch hatte mich für die Rolle "All Features" entschieden. Anschließend stellte sich mir die IDE in folgendem Gewande da:



Anhand der Literatur, die ich mir besorgt hatte, wusste ich, dass ich als erstes einmal eine Applikation erstellen musste. Also startete ich mit der Auswahl von „New Application“. *ADF* ist eine deklarative Entwicklungs-Umgebung. Das bedeutet unter anderem, dass es hier eine ganze Menge hilfreicher „Wizards“ gibt. Wenn man zum Beispiel eine neue Applikation erstellt, wird alles notwendige der Reihe nach abgefragt. Bestimmte Entscheidungen sollten allerdings vorher durchdacht werden. So macht es Sinn, nicht den Vorschlag des Wizards zu übernehmen, die Applikation in ein separates Verzeichnis zu legen. Beim Anlegen der Applikation wurde ich durch

den Wizard auch gefragt, wie mein erstes Projekt heißen soll. Das war neu für mich. In meiner gewohnten *Forms*-Umgebung hatte ich bisher immer ein Modul, welches benannt werden musste.

Hier mal ein Überblick der Nutzeroberfläche von *JDeveloper*:



Im Fenster ist eine bestehende Applikation geöffnet. Im Vergleich zum *Formsbuilder* muss ich sagen, dass ich einige Sachen gefunden habe, die hier ähnlich sind und den Einstieg für mich erleichtert haben.

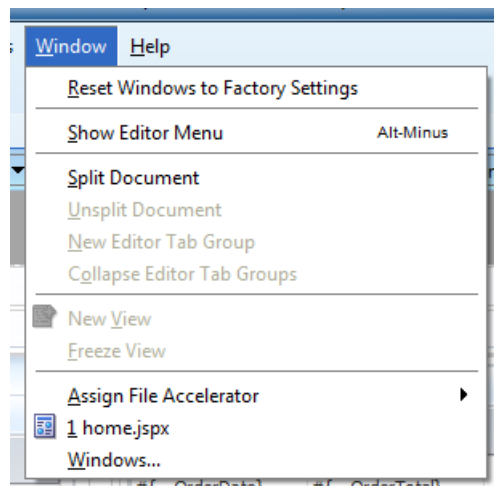
Da ist zum Beispiel ein Navigator, der bei *Forms* die Struktur der Maske darstellt - hier ebenfalls auf der linken Seite. Der *Application Navigator* zeigt eine datei-basierte Sicht auf die Struktur der Applikation. Durch Doppelklick auf einzelne Dateien werden diese in der Mitte, d.h. im zentralen Arbeitsbereich geöffnet. Bisher ist vieles der Entwicklung von *Forms* ziemlich ähnlich.

Im nächsten Bereich, der *Structure Pane* (Strukturbereich), wird die jeweilige Struktur dargestellt, ausgehend von der aktuell mit dem Cursor markierten Komponente der Applikation. Das wiederum kam mir aus *Oracle Reports* bekannt vor. Auch dort habe ich eine strukturelle Sicht auf den Report.

Ebenso wieder erkannt habe ich die *Component Palette*. Allerdings habe ich in *JDeveloper* weitaus mehr Komponenten zur Verfügung, als in meiner alten Entwicklungsumgebung - ca. 150 verschiedene.

Am zweiten Tag habe ich mich ausschließlich mit der IDE beschäftigt. Auf den ersten Blick machte sie einen recht unaufgeräumten Eindruck, aber ich lernte bald die Möglichkeiten kennen, wie man alles an die eigenen Bedürfnisse anpassen kann.

Zum Schluss noch ein Tipp: Das wichtigste Menü an diesem Tag war für mich das folgende:



Warum? Jedes Mal, wenn ich meine Sichten so verschoben hatte, dass ich nicht mehr durchsah, half mir ein "Reset", die ursprüngliche Anordnung wiederherzustellen.

Dritter Tag: Meine Daten

Nachdem ich mich mit den Grundlagen der IDE beschäftigt hatte, wurde es Zeit, einmal eine kleine Anwendung zu bauen. Dazu nahm ich das folgende Buch zu Hilfe:

Grant, Ronald:

Quick Start Guide to Oracle Fusion Development: Oracle JDeveloper and Oracle ADF (2010)

Die ersten Schritte begannen damit, wie man eine Datenbasis erstellt. Bei *Oracle Forms* heißt dieser Bereich "Data Block Wizard", im *JDeveloper* nimmt man dafür den Wizard zum Erstellen von ADFbc Komponenten. Zunächst musste ich eine Datenbankverbindung erstellen, um dann die Tabellen aus der Datenbank auszuwählen, mit denen ich arbeiten wollte. Neu an dieser Stelle war für mich, dass ich diese Prozedur gleich noch einmal wiederholen musste, um *nur-lesende Objekte* zu erstellen. Aber sehr schnell hatte ich alle benötigten „Datenblöcke“ zusammen. Außerdem konnte ich einzelne Eigenschaften wie gewohnt verändern. Allerdings ist die Anzahl und Bezeichnung dieser „Properties“ anders als bei *Forms*, in *JDeveloper* lehnen sie sich doch eher an die Java Schreibweise an.

Mir hat am dritten Tag besonders gefallen, dass ich zwischen meiner alten Entwicklungsumgebung und *JDeveloper* viele Gemeinsamkeiten gefunden habe. So konnte ich, wie gewohnt, Business Components und Attribute (Felder) in meinen „Blöcken“ hinzufügen. Ebenso gestaltete sich das Hinterlegen von Labels, Hints, Formaten und Prüfungen ähnlich wie bei *Forms*.

Vierter Tag: Meine Leinwände

Nachdem ich meine Daten bereitgestellt hatte, kam jetzt die Aufgabe auf mich zu, eine Webanwendung zu erstellen. Ohne weitere Vorkenntnisse war das am Anfang komplizierter, als ich ursprünglich vermutet hatte. Es gibt sehr viele Komponenten, die man auf einem Panel (auch *Leinwand* genannt) platzieren kann. Es war daher hilfreich, dass ich zwischendurch mal einen Tagesworkshop besucht hatte, um einen kleinen Einblick in *User interfaces mit ADF* zu bekommen. Aber wenn die ersten Hürden einmal genommen sind, macht es richtig Spaß, Objekte einer Anwendung auf der *Leinwand* zu platzieren. Für mich als *Forms*-Entwickler stellte sich das

dann genauso einfach da, wie ich es von früher gewohnt war. Ebenfalls konnte ich die notwendigen Attribute (Properties) gut über den Inspektor modifizieren. Nachdem die Objekte zusammengebaut waren konnte ich endlich den „Run“ Button drücken und meine Anwendung wurde ausgeführt. Toll.

Mein erster Versuch war kein aufwändiges Programm, sondern nur ein einfaches Master-Detail-Detail. Aber ich hatte es innerhalb kurzer Zeit geschafft, eine lauffähige Anwendung zu erstellen, mit der man Daten suchen, einfügen, ändern und löschen kann. Dafür hatte ich keine einzige Zeile Java-Code schreiben müssen.

Fazit

Mit *ADF* und dem *JDeveloper* wurde ich innerhalb kürzester Zeit in die Lage versetzt, erste Erfolge zu erzielen, und das obwohl ich kein Java-Entwickler bin. Ich habe zum Glück sehr viele Elemente wieder erkannt, die mir schon von *Oracle Forms* oder *Oracle Reports* bekannt waren. *JDeveloper* liefert Funktionen, mit denen man durch ein paar Mausklicks einfache Masken erstellen kann, aber für komplexe Projekte reicht das natürlich nicht aus. Um von hier aus weiter zu kommen, sollte man die Unterstützung durch einen erfahrenen Entwickler suchen. Ich jedenfalls freue mich schon auf mein nächstes Projekt!

Die Seitenzahl wird von uns eingefügt!

Bitte fügen Sie Ihre Kontaktadresse hinzu.

Kontaktadresse:

Jan-Peter Timmermann
Trivadis GmbH
Paul-Dessau-Str. 6, Gebäude P
D-22761 Hamburg

Telefon: +49 (0) 40-248591 30
Fax: +49 (0) 40-248591 59
E-Mail jan-peter.timmermann@trivadis.com
Internet: www.trivadis.com