

# Aussehen ist alles

## Modernisierung von Forms durch neues Look&Feel

Dr. Gudrun Pabst  
Trivadis  
München

### Schlüsselworte:

Oracle Forms, Pluggable Java Components, PJC

### Einleitung

Eine häufig geäußerte Kritik an Forms-Anwendungen ist, dass die Oberflächengestaltung sachlich, nüchtern und im Vergleich mit manchen anderen Web-Anwendungen "langweilig" ist. Mit Hilfe von Pluggable Java Components kann eine bestehende Forms-Anwendung mit einem neuen Aussehen versehen werden.

Im Folgenden wird gezeigt, wie derartige Pluggable Java Components (PJC) erstellt werden. Um die Verwendung der erstellten PJC in Forms zu demonstrieren, werden sie zunächst in eine (leere) Forms-Maske eingebunden. Abschließend werden Möglichkeiten besprochen, wie eine bestehende Anwendung auf das neue Layout umgestellt werden kann.

### Ausgangssituation

Viele Forms-Anwendungen bestehen seit langer Zeit. Auch wenn sie auf neue Forms-Versionen umgestellt und ins Web gebracht worden sind, ist im allgemeinen am Layout nichts geändert worden:

Oracle Application Server Forms ...

Abt. 1

Schulungstermine

**Schulungstermine**

Schulung: 202  
von: 30.05.2010 13:00  
Ort: Stuttgart-Vaihingen  
Referent: Frau Erber  
Status: geplant

Titel: Schulung AV 4711  
bis: 30.05.2010 17:30  
Externer Referent?:   
Testversion:

**Teilnehmer**

Stamnummer	Nachname	Vorname	abgezeichnet von	Bemerkung

Record: 1/?

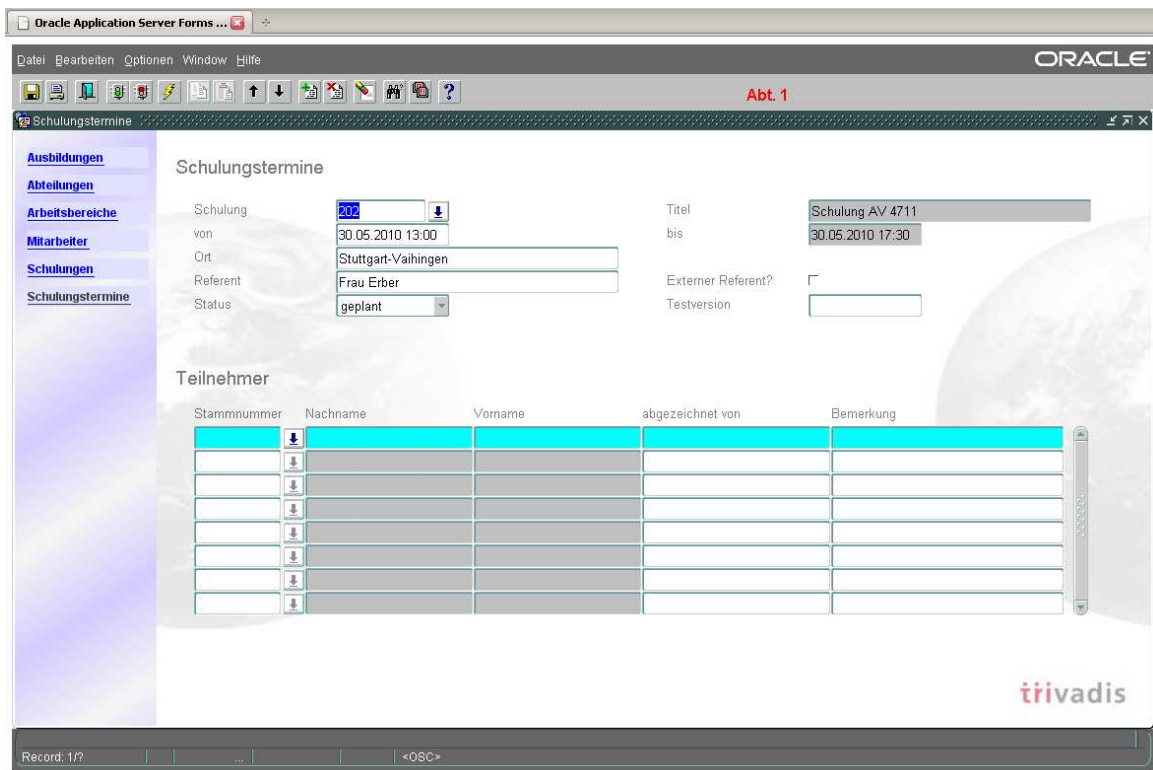
<080>

ORACLE

trivadis

Im ersten Schritt muss die Anwendung analysiert werden, um festzustellen, welche unterschiedlichen Layouttypen vorhanden sind. Dann muss ein konsistentes neues Layout entwickelt werden, das jeden vorhandenen Typen abdeckt. Abschließend wird geprüft, welche Forms-Objekte angepasst werden müssen, damit das neue Layout umgesetzt werden kann. Anregungen für ein neues Layout bietet z.B. das „Oracle Forms Look & Feel project“ von Francois Degrelle.

Das hier verwendete neue Layout könnte natürlich noch weiter verfeinert werden. Dieses Layout benutzt nur einen angepassten Canvas für den Hintergrund des Hauptbereichs und der Navigationsleiste sowie JavaBeans zur Umsetzung von Hyperlinks:



## Pluggable Java Components (PJC)

Pluggable Java Components erlauben die Anpassung und Erweiterung von Forms-Objekten. Da Forms im Web als Java-Applet läuft, sind alle Forms-Objekte durch Java-Klassen realisiert. Eine Pluggable Java Component ist eine Java-Klasse, die die von Forms vorgegebene Java-Klasse erweitert und abändert. Eine PJC muss dabei eine JavaBean sein, d.h. der JavaBean-Spezifikation entsprechen.

Die Spezifikation von JavaBeans legt fest: Eine JavaBean kann folgende Strukturelemente enthalten:

- Properties  
Diese sind gekennzeichnet dadurch, dass die JavaBean zwei Methoden  

```
set<Property> ( <PropertyType> value );
```

 und  

```
<PropertyType> get<Property>;
```

 enthält.
- Indexed Properties  
Eine Indexed Property liegt vor, wenn die JavaBean zwei Methodenpaare `set<Property>` und `get<Property>` für die gesamte Property und für einen einzelnen Indexwert besitzt:  

```
set<Property> ( int index , <PropertyType> value );
```

```
<PropertyType> get<Property> ( int index );  
set<Property> ( <PropertyType> values[] );  
<PropertyType>[] get<Property>;
```

➤ **Methoden**

Hier ist jede Java-Methode möglich.

➤ **Events**

Listener überwachen das Auftreten der Events in der JavaBean und geben eine Information über den Event an interessierte Benutzer der JavaBean weiter.

Der Enhanced JavaBean Support von Forms kann diese Elemente ansprechen.

## **PJCs für das neue Layout**

Das angestrebte neue Layout verwendet zwei PJCs: eine PJC für die Canvas-Gestaltung und eine PJC für die Hyperlinks.

Für den Canvas wird eine Klasse definiert, die auf der Klasse `oracle.forms.ui.VBean` beruht:

```
import oracle.forms.ui.VBean;  
public class AlternativCanvas extends VBean {
```

Diese Klasse fügt dem Canvas ein neues JPanel (`javax.swing.JPanel`) hinzu, das entweder das Hintergrundbild anzeigt oder den Farbverlauf. Mit der Methode `paint` wird dafür gesorgt, dass die Ergebnisse am Bildschirm angezeigt werden:

```
vJPanel = new JPanel () {  
    public void paint ( Graphics pGraphics ) {  
        // Code zum Anzeigen des Bildes oder des Farbverlaufs  
    }  
}
```

Zum Setzen des Bildes bzw. der Farben für den Gradienten werden `set-` und `get-` Methoden definiert, die dann von Forms aus als Properties angesprochen werden können:

```
public void setGradientColors ( String value ) { /* ... Code ... */ }  
public String getGradientColors ( ) { /* ... Code ... */ }  
public void setBackgroundImage ( String pImageName ) { /* ... Code ... */ }  
public String getBackgroundImage ( ) { /* ... Code ... */ }
```

Zur Umsetzung des Hyperlinks wird die Klasse verwendet, die Oracle in den Demos zu Forms mitliefert.

## **Einbinden in Forms**

Die erstellten Klassen werden zusammen mit den Hintergrundbildern in eine Jar-Datei gepackt. Diese sollte zusätzlich noch signiert werden. Die Jar-Datei wird dann auf dem Forms-Server im Java-Verzeichnis zur Verfügung gestellt. Die Konfiguration wird so angepasst, dass die neue Jar-Datei beim Aufruf der Forms-Module verwendet wird.

Zum Einbinden der PJCs in Forms werden Items vom Typ „Bean Area“ erstellt. Diese Items müssen einem Canvas zugeordnet und sichtbar sein (die Größe darf allerdings 1 Pixel in Höhe und Breite betragen). Mit der Package `fbean` werden die PJCs verwaltet.

Zum gewünschten Zeitpunkt, z.B. im WHEN-NEW-FORM-INSTANCE-Trigger, wird die Bean Area (Item MAIN\_BEAN im Block CTL) mit der JavaBean verbunden:

```
fbean.register_bean ( 'CTL.MAIN_BEAN' , 1 , 'pjc2010.AlternativCanvas' );
```

Dies löst die Initialisierung der JavaBean aus. Anschließend können Properties der JavaBean gesetzt werden. Durch die Definition der Methoden setBackgroundImage und getImage verfügt die JavaBean über die Property backgroundImage, die von Forms aus mittels

```
fbean.set_property ( 'CTL.MAIN_BEAN' , 1 , 'backgroundImage' ,  
                    'background.jpg' );
```

gesetzt werden kann.

Für die Navigationsleiste im linken Bereich wird ein neuer Canvas vom Typ “Vertical Toolbar” erstellt sowie ein Block mit so vielen Items vom Typ “Bean Area” wie Links benötigt werden. Die Items sind so groß, dass die Link-Beschriftungen angezeigt werden können. Die Items werden beim Start mit der Hyperlink-Klasse verbunden und mit dem entsprechenden Label versehen. Zusätzlich wird der Block (oder jedes Item) mit einem WHEN-CUSTOM-ITEM-EVENT-Trigger versehen, um die Aktion zu hinterlegen, die beim Klicken eines Hyperlinks ausgelöst wird.

### **Umstellen einer bestehenden Anwendung**

Nach dem Programmieren der PJC's und der PL/SQL-Routinen zu ihrer Verwendung muss jetzt die bestehende Anwendung umgestellt werden. Je nach Aufbau der Anwendung ist dies einfach zu lösen oder mit Aufwand verbunden.

Die hier durchgeführte Anpassung wird durch folgende Forms-Objekte umgesetzt:

- ein vertikaler Toolbar-Canvas
- ein Block und mehrere Bean Areas für die Hyperlinks
- eine Bean Area für die Gestaltung des Haupt-Canvas; diese Bean Area muss auf dem Haupt-Canvas liegen
- eine Bean Area für die Gestaltung des neuen Toolbar Canvas

Diese Objekte werden in einer Object Group zusammengefasst.

Gibt es für Referenzen bereits eine Object Library, wird die Object Group in der Object Library gespeichert. Ist sogar eine Object Group vorhanden, die in jedem betroffenen Forms-Modul referenziert ist, werden die neuen Objekte zu dieser Object Group hinzugefügt und in der Object Library gespeichert. Nach dem nächsten Speichern bzw. Kompilieren stehen die Objekte damit in allen Forms-Modulen zur Verfügung.

Weitere Objekte, die angepasst werden müssen, können ebenfalls über die Object Library geändert werden, wenn sie auf Vorlagen beruhen. Im hier vorgestellten Fall muss die Breite des Hauptfensters geändert werden und die Hintergrundfarbe des Haupt-Canvas auf weiß umgestellt werden.

Verwendet die Anwendung eine Library-Routine im WHEN-NEW-FORM-INSTANCE-Trigger (bzw. dem Trigger, der für die Initialisierung gewählt wurde), so wird einfach die Routine um den Initialisierungscode erweitert.

Meist lassen sich damit nicht alle Forms-Module abdecken; Spezialfälle müssen unter Umständen noch manuell nachbearbeitet werden.

Verwendet die Anwendung keine Referenzen, so empfiehlt sich, für die neu erstellte Object Group dennoch eine Object Library anzulegen und die neuen Objekte aus dieser Object Library zu referenzieren. Ebenso ist es sinnvoll, den Initialisierungscode in einer Library abzulegen.

Um die Object Group dann in allen Forms-Modulen zu referenzieren und die neue Library-Routine aufzurufen, können die Module entweder mit der Java-API bearbeitet werden. Alternativ können die Module in XML umgewandelt, angepasst und zurückkonvertiert werden. Beide Möglichkeiten setzen aber voraus, dass es einen automatisierbaren Weg gibt, die Objekte zu bestimmen, die angepasst werden müssen.

Gibt es in der Anwendung keine Möglichkeit, die anzupassenden Objekte automatisch zu identifizieren, müssen die Forms-Module einzeln manuell überarbeitet werden. Mit der Verwendung einer Object Library und Code aus einer PL/SQL-Library lassen sich aber auch in diesem Fall die Arbeiten mit begrenztem Aufwand durchführen.

#### **Links:**

JavaBean-Spezifikation:

<http://www.oracle.com/technetwork/java/javase/documentation/spec-136004.html>

Sammlungen von PJC's:

<http://forms.pjc.bean.over-blog.com/>

Umfangreiches Look&Feel-Projekt:

[http://fdtool.free.fr/LAF/doc/Oracle\\_Forms\\_Look\\_and\\_Feel\\_project.htm](http://fdtool.free.fr/LAF/doc/Oracle_Forms_Look_and_Feel_project.htm)

#### **Kontaktadresse:**

##### **Dr. Gudrun Pabst**

Trivadis

Lehrer-Wirth-Straße 4

D-81829 München

Telefon: +49 (0) 89-99 27 59 30

Fax: +49 (0) 89-99 27 59 59

E-Mail [gudrun.pabst@trivadis.com](mailto:gudrun.pabst@trivadis.com)

Internet: [www.trivadis.com](http://www.trivadis.com)