

■ ■ ■ Modernisierung von Forms durch neues Look&Feel



Agenda



- Definition des neuen Layout
- Pluggable Java Components (PJC's)
- Erstellen einer PJC
- Einbinden in Forms
- Anpassen einer bestehenden Anwendung

Definition des neuen Layout



- Analyse der bestehenden Anwendung:
 - Welche Typen von Forms-Masken gibt es?
 - Wie soll das Layout zukünftig aussehen?
- Festlegen der benötigten Layout-Varianten
- Ermitteln der Forms-Objekte, die für das neue Layout angepasst werden müssen
- Achtung: Machbarkeit – nicht alles ist umsetzbar!

Definition des neuen Layout



- Bisheriges Aussehen:

Oracle Application Server Forms ...

Datei Bearbeiten Optionen Window Hilfe

ORACLE

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

trivadis

Record: 1/?

Definition des neuen Layout

- ■ ■ ■
- Gewünschtes neues Aussehen:

The screenshot displays the Oracle Application Server Forms interface for 'Schulungstermine'. The window title is 'Oracle Application Server Forms ...'. The menu bar includes 'Datei', 'Bearbeiten', 'Optionen', 'Window', and 'Hilfe'. The toolbar contains various icons for file operations and navigation. The main content area is divided into two sections: 'Schulungstermine' and 'Teilnehmer'.

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
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Record: 1/?

Definition des neuen Layout



- Das gewünschte neue Layout könnte natürlich noch verfeinert werden

- Das neue Layout umfasst folgende Änderungen:
 - Haupt-Canvas mit Hintergrundbild
 - Neuer vertikaler Toolbar-Canvas mit Farbverlauf
 - Hyperlink-Objekte

- Anzupassende Forms-Objekte:
 - Canvas
 - Bean Area

Agenda



Daten sind
immer im Spiel.

- Definition des neuen Layout
- Pluggable Java Components (PJC's)
- Erstellen einer PJC
- Einbinden in Forms
- Anpassen einer bestehenden Anwendung

Pluggable Java Components (PJC)



- Sie erlauben die Anpassung und Erweiterung von Forms-Objekten
- PJC sind Java Beans, d.h. sie müssen der Java Bean – Spezifikation entsprechen
- Eine JavaBean umfasst:
 - Properties
d.h. `get<Property>-`, `set<Property>-`Methoden
 - Indexed Properties
d.h. `get<Property>-`, `set<Property>-`Methoden für die gesamte Property und für einen einzelnen Index
 - Methoden
 - Events
- Der Enhanced JavaBean-Support von Forms kann diese Elemente ansprechen.

Agenda



Daten sind
immer im Spiel.

- Definition des neuen Layout
- Pluggable Java Components (PJC's)
- Erstellen einer PJC
- Einbinden in Forms
- Anpassen einer bestehenden Anwendung

Erstellen einer PJC



- Für das neue Layout: PJC für den grafisch gestalteten Canvas
- Neue Java-Klasse basierend auf `oracle.forms.ui.VBean`:

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

- Grafik des Canvas nicht direkt modifizierbar, daher:
neues JPanel mit der gewünschten Grafik hinzufügen

```
vJPanel = new JPanel () {  
    public void paint ( Graphics pGraphics ) {  
        // Code für die gewünschte grafische Gestaltung  
    }  
}
```

Erstellen einer PJC



- set- und get-Methoden:
- für das Setzen der Farben für den Farbverlauf aus Forms:

```
public void    setGradientColors (String value) {...}  
public String getGradientColors () {...}
```

- für das Setzen des Hintergrundbildes:

```
public void    setBackgroundImage (String pImage) {...}  
public String  setBackgroundImage () {...}
```

- Für das Umsetzen des Hyperlink: Klasse Hyperlink.java aus den Oracle Demos

Agenda



Daten sind
immer im Spiel.

- Definition des neuen Layout
- Pluggable Java Components (PJC's)
- Erstellen einer PJC
- Einbinden in Forms
- Anpassen einer bestehenden Anwendung

Einbinden in Forms



Vorarbeiten:

- Packen der Java-Klassen in ein Jar-File
- Signieren des Jar-Files
- Ablegen des Jar-Files auf dem Application Server
- Einbinden des Jar-Files in die Konfiguration des Projekts in der Datei formsweb.cfg:

```
archive=frmall.jar, ..., pjc2010.jar
```

Einbinden in Forms



Einbinden in Forms:

- Erstellen eines Items
 - vom Typ Bean Area
 - auf dem anzupassenden Canvas
 - sichtbar (!), aber mit Breite und Höhe je 1 Pixel
- Verwaltung mit der Package fbean:
 - Verknüpfung des Items mit der Java-Klasse und dadurch Initialisierung z.B. im WHEN-NEW-FORMS-INSTANCE-Trigger:

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

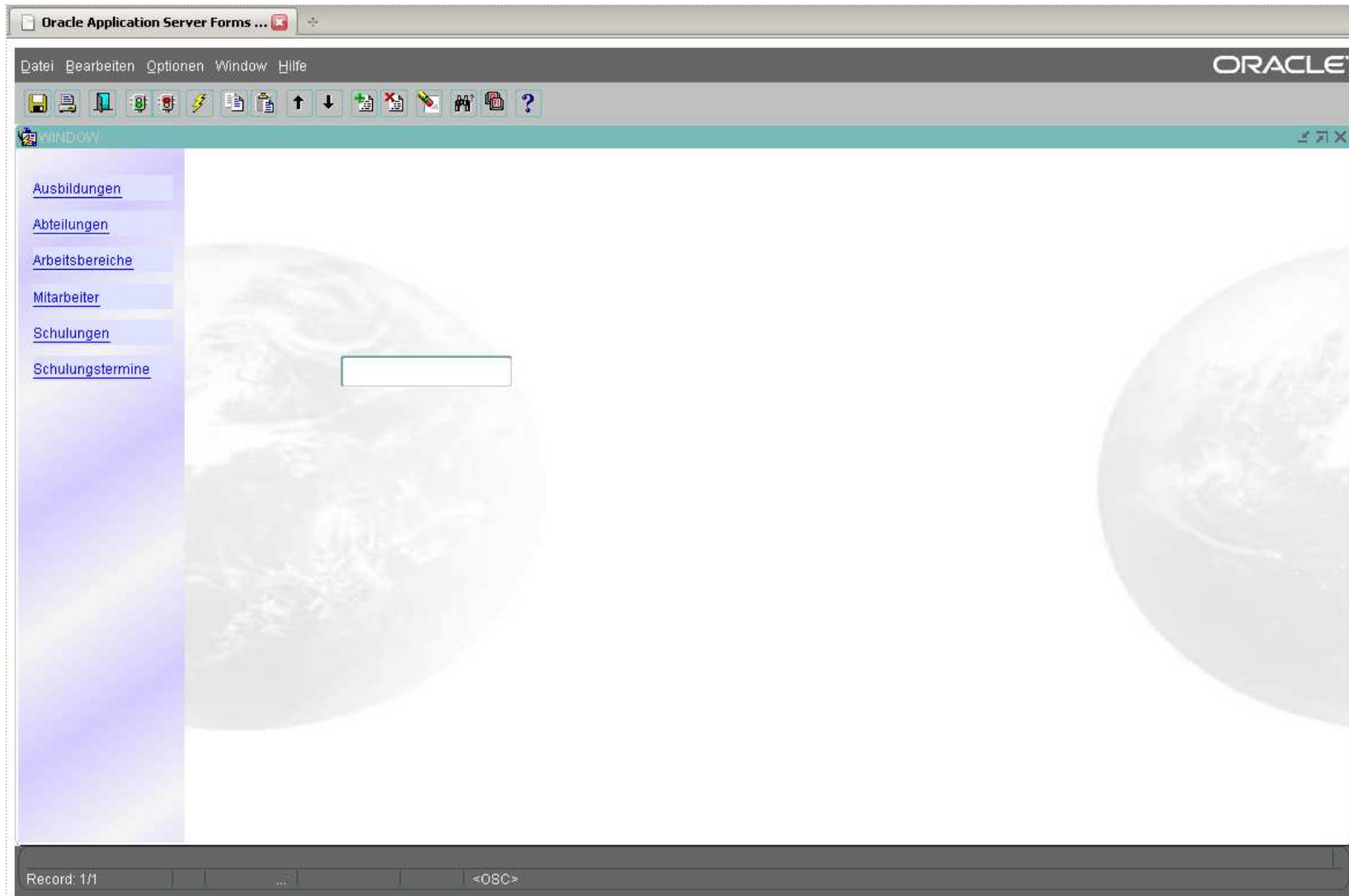
- Anschließend Setzen des Hintergrundbildes:

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

Einbinden in Forms



- Leere Vorlage:



Agenda



Daten sind
immer im Spiel.

- Definition des neuen Layout
- Pluggable Java Components (PJC's)
- Erstellen einer PJC
- Einbinden in Forms
- Anpassen einer bestehenden Anwendung

Anpassen einer bestehenden Anwendung



Vorarbeiten:

- Hinzufügen der benötigten Objekte zu einer zentralen Object Group (falls vorhanden):
 - vertikaler Toolbar-Canvas
 - Block und Bean Areas für die Hyperlinks
 - Bean Area für Haupt-Canvas
 - Bean Area für neuen Toolbar Canvas

- Ablegen des Initialisierungscode in einer PL/SQL-Library

Anpassen einer bestehenden Anwendung



Einfache Anpassung, wenn

- Anwendung basiert auf referenzierten Objekten
- zentrale, überall verwendete Object Group vorhanden
- gemeinsamer Initialisierungscode aller Masken existiert

Umstellung der Anwendung durch

- Hinzufügen der neuen Objekte zur zentralen Object Group
- ggf. Anpassen weiterer Vorlage-Objekte, hier:
 - Window-Breite vergrößern
 - vertikale Toolbar zum Window zuordnen
 - Canvas-Hintergrund des Haupt-Canvas weiß
- Erweitern des Initialisierungscode um neue Prozedur
- Kompilieren aller Forms-Module

Anpassen einer bestehenden Anwendung



Schwieriger

- Anwendung basiert nicht auf referenzierten Objekten
- keine zentrale Object Group oder
- keine erweiterbare Initialisierungsroutine vorhanden

Umstellung durch

- automatisiertes Anpassen
 - über die Java API
 - über die XML-Darstellung der Forms-Module
- manuelles Einfügen der Referenzen
 - bei Verwendung von Object Groups und PL/SQL-Libraries nicht schwierig, aber zeitaufwendig

■ ■ ■ Fragen?



?

www.trivadis.com

trivadis

makes **IT** easier. ■ ■ ■

