

Nekad i sad - od Forms-a do ADF-a

Patrik Franković

Roland Miklić

Sektor za informatiku i telekomunikacije

Sadržaj

- Forms vs. ADF
- Kako započeti
- Model i View Controller
- jsp, jsff
- Tips & Tricks

ADF ukratko

- Skrivena kompleksnost tehnologije
 - Vizualni i deklarativni razvoj
 - ADF se brine o “instalacijama”
 - Metadata, a ne programski kod
- Tehnologija koju je Oracle izabrao za svoje buduće poslovne aplikacije
- Bazirana na standardima i open source-u
- ADF BC – prilično jednostavno za Forms developere
- Java je neizbježna

Forms vs. ADF



Forms	ADF
Konekcija na bazu	Konekcija na bazu
Block = table (izvor podataka)	View = table (izvor podataka)
Kešira podatke unutar Forms record manager	Kešira podatke unutar entity objects
Brine se o transakcijama	Brine se o transakcijama
Primjenjuje bazna pravila(database constraints)	Primjenjuje bazna pravila(database constraints)
Mogućnost primjene deklarativne validacije	Mogućnost primjene deklarativne validacije
Koristi trigger za primjenu korisničkih poslovnih pravila	Koristi evente za primjenu korisničkih poslovnih pravila
Mogućnost ugradnje raznih built-in-ova	Mogućnost ugradnje raznih built-in-ova
Definiranje svojstava za pojedine elemente	Definiranje svojstava za pojedine elemente
Razne vizualne komponente	Razne vizualne komponente

Forms 2 ADF

Form

Task
Flow

Canvas

JspX +
jsff

Block

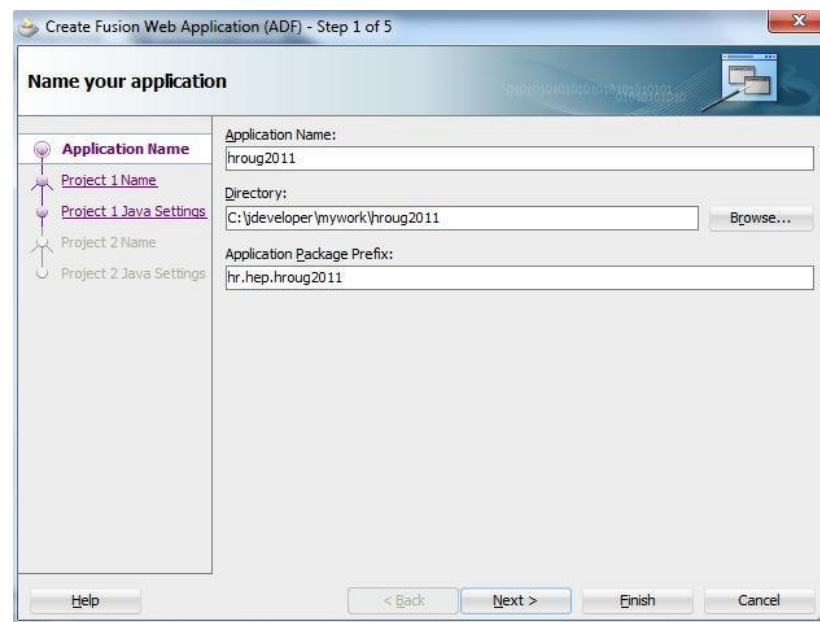
View

Trigger

Java +
Groovy

Svaki početak je težak

- Good ol' Forms
- HTML i web dizajn? Čuo sam da susjedov mali...
- Java? Ne, hvala!
- MVC – modeliraj – vidi - kontroliraj



Planiranje je važno

hroug2011 Overview

Checklist

- Java Files
- Page Flows
- Managed Beans
- Web Tier
- Business Components
- Binding Files
- Offline Database

Fusion Web Application Quick Start Checklist

Create the application by following step-by-step instructions describing how to build Fusion Web Applications according to Oracle best practice recommendations.

Show All | Hide All

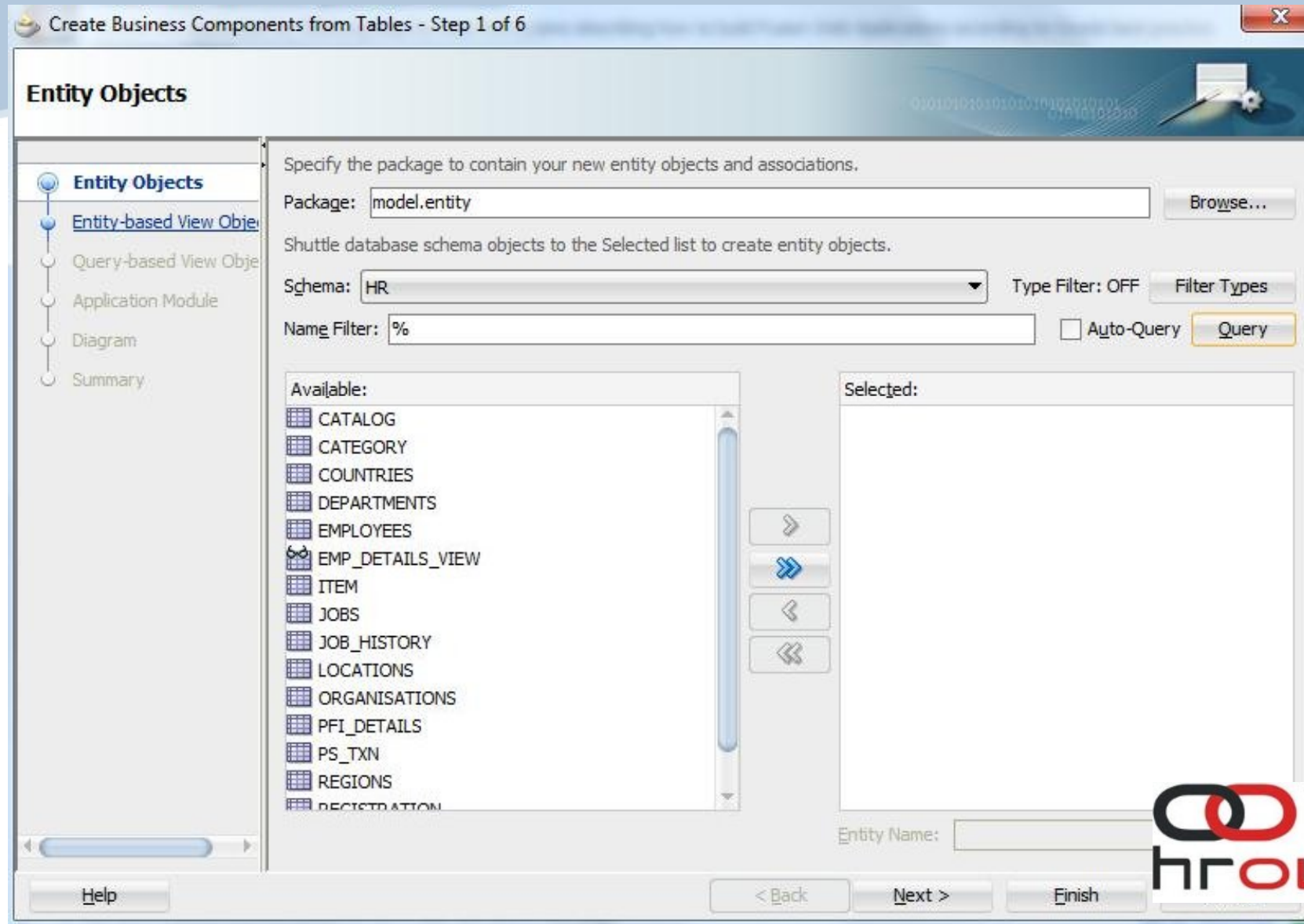
1	▶ Plan Your Application	<input type="checkbox"/> Not Started
2	▶ Connect to a Database	<input type="checkbox"/> Not Started
3	▶ Build Business Services	<input type="checkbox"/> Not Started
4	▶ Design Application Flow	<input type="checkbox"/> Not Started
5	▶ Design Pages	<input type="checkbox"/> Not Started
6	▶ Add Common Components (Lookups, Search and Menus)	<input type="checkbox"/> Not Started
7	▶ Implement Business Logic	<input type="checkbox"/> Not Started
8	▶ Secure Your Application	<input type="checkbox"/> Not Started
9	▶ Internationalize Your Application	<input type="checkbox"/> Not Started
10	▶ Debug and Test Your Application	<input type="checkbox"/> Not Started
11	▶ Package and Deploy Your Application	<input type="checkbox"/> Not Started
12	▶ SOA-Enable Your Application	<input type="checkbox"/> Not Started

Dobar **plan** danas **bolji** je nego savršen **plan** sutra.

General Paton



Kvisko je zakon!



2 projects to rule them all

hroug2011

Projects

Model

- Application Sources
- model
- model.entity
- model.modules
- model.uiview

ViewController

- Application Sources
- Web Content

• Tables

- Entity Objects
- View Objects
- View Links / Associations
- Query-Based Blocks -> View Objects
- LOVs -> read-only View Objects

• Forms -> Taskflows

• Windows & Canvases

- Windows -> JSPX pages
- Canvases -> Fragments, Panels

BC Tester

The screenshot displays the Oracle ADF Model Tester interface. On the left, a project tree shows the structure of 'Hroug2011AM', including 'Application Sources', 'model', 'model.entity', and 'model.modules'. The main window shows the 'EmpDeptFkLink1' entity with a data table. The table has columns for EmployeeId, FirstName, LastName, Email, PhoneNum..., and HireDate. The first row of data is highlighted in blue.

EmployeeId	FirstName	LastName	Email	PhoneNum...	HireDate
200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17 ...

Name: Hroug2011AM.EmpDeptFkLink1 Definition: model.uiview.EmpDeptFkLink

Muke sa sekvencom

Attributes

Set Source Ord

Entity attributes can be based upon columns in the schema object or can be based upon transient values.

Name	Type	Column Name	Column Type	Extends
DepartmentId	oracle.jbo.domain.DB...	DEPARTMENT_ID	NUMBER(4, 0)	
DepartmentName	String	DEPARTMENT_NAME	VARCHAR2(30)	
ManagerId	Integer	MANAGER_ID	NUMBER(6, 0)	
LocationId	Integer	LOCATION_ID	NUMBER(4, 0)	

Details | UI Hints | Validation Rules | Security | Dependencies | Custom Properties

Name : DepartmentId

Display Name: Department Id

Description:

Type : DBSequence

Property Set: <None>

Default Value

Literal Expression SQL

Updatable: While New

Persistent Transient

Mandatory Refresh on Insert

Primary Key Refresh on Update

Queryable Change Indicator

Precision Rule

Track Change History: none

Groovy izraz za default value za slučaj da nema triggera u bazi:

`(new oracle.jbo.server.SequenceImpl(„DEP_SEQ“, object.getDBTransaction())).getSequenceNumber()`

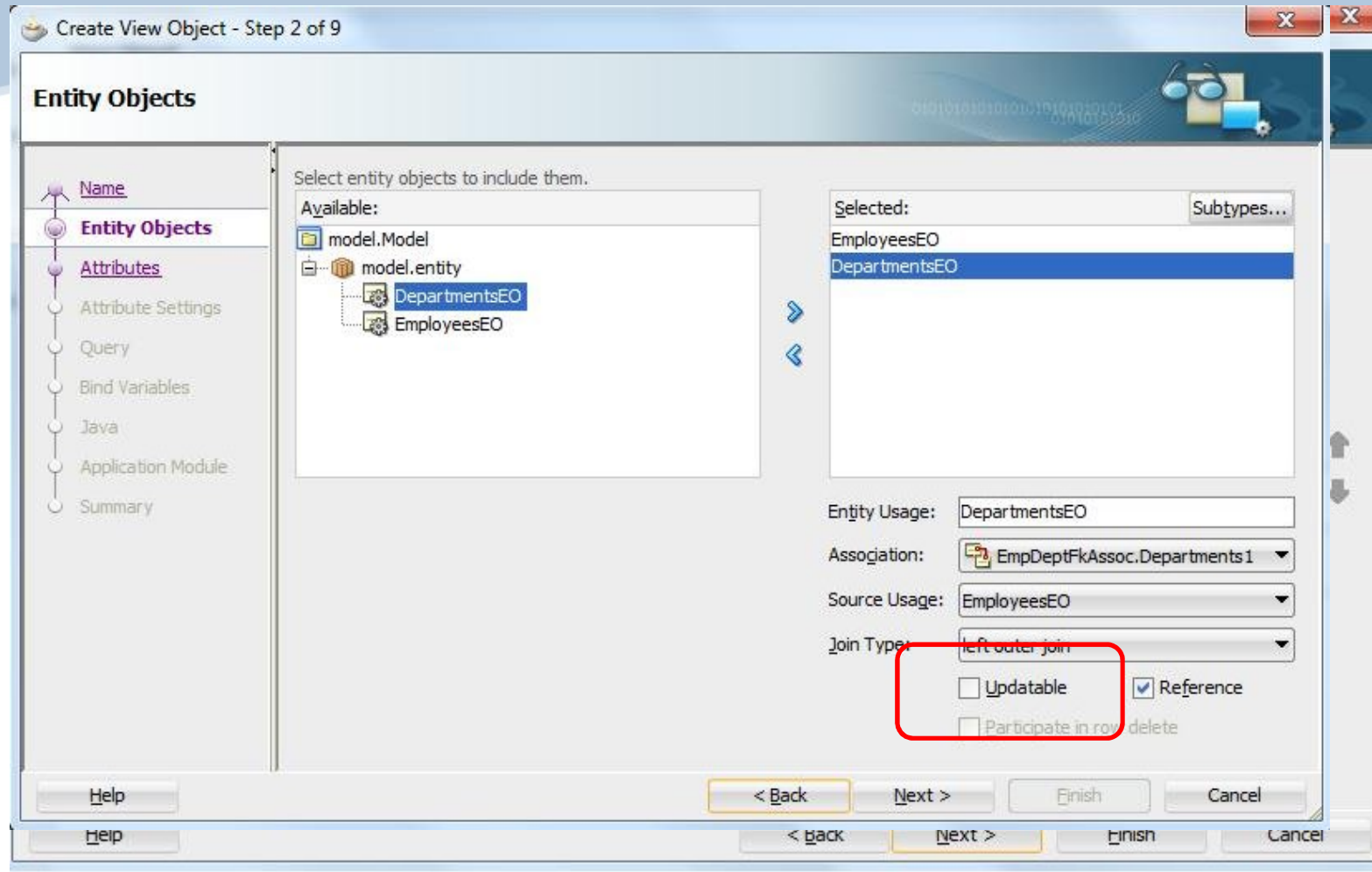
I love LOV

The screenshot shows the Oracle ADF IDE interface. The 'Attributes' panel is open, showing a list of attributes for an entity. The 'DepartmentId' attribute is selected. The 'Edit List of Values' dialog box is open, showing the configuration for the 'LOV_DepartmentId' list. The 'List Data Source' is set to 'DepartmentsVO1' and the 'List Attribute' is set to 'DepartmentId'. The 'List Return Values' table shows a mapping from 'DepartmentId' to 'DepartmentId'. A red arrow points from the 'View Accessors' menu item in the background to a callout box labeled 'View accessor'.

Name	List Data Source	List Attribute
LOV_DepartmentId	DepartmentsVO1	DepartmentId

View accessor

Post Post-change razdoblje



Stay tuned

Tuning

Enter tuning parameters for this View, to control SQL execution and how data is fetched from the database.

Retrieve from the Database

All Rows Only up to row number

in Batches of:

As Needed All at Once

At Most One Row

No Rows (i.e. used only for inserting new rows)

Query Optimizer Hint:
e.g FIRST_ROWS, ALL_ROWS, etc.

Fill Last Page of Rows when Paging through Rowset

Passivate State (e.g. Current Row, Bind Values, etc.)

Including All Transient Values

Retain View Link Accessor Rowset

Access Mode:

Range Size:

Naročito korisno kod LOV view objekata s velikim brojem podatka.

Where uvjet ili View Criteria

The image shows a screenshot of the Oracle SQL Developer interface. On the left, a query is displayed in the 'Query' tab, which retrieves data for a view object from the 'EMPLOYEES' table. The query is as follows:

```
SELECT EmployeesEO.EMPLOYEE_ID,  
       EmployeesEO.FIRST_NAME,  
       EmployeesEO.LAST_NAME,  
       EmployeesEO.EMAIL,  
       EmployeesEO.PHONE_NUMBER,  
       EmployeesEO.HIRE_DATE,  
       EmployeesEO.JOB_ID,  
       EmployeesEO.SALARY,  
       EmployeesEO.COMMISSION_PCT,  
       EmployeesEO.MANAGER_ID,  
       EmployeesEO.DEPARTMENT_ID  
FROM EMPLOYEES EmployeesEO
```

Below the query, the 'Bind Variables' and 'View Criteria' sections are visible. The 'View Criteria' section is currently empty.

The main window is the 'Create View Criteria' dialog box. It is titled 'Create View Criteria' and has a 'Criteria Name' field set to 'EmployeesbyFirstName'. The 'Query Execution Mode' is set to 'Database'. The dialog has two tabs: 'Criteria Definition' and 'UI Hints'. The 'Criteria Definition' tab is active, showing a tree view of the criteria definition. The tree view shows a 'View Criteria' named 'EmployeesVOCriteria' containing a 'Group' with a single 'Criteria Item' named 'FirstName = :vIme'. The 'View Object Where Clause' field contains the following SQL expression: `(((EmployeesEO.FIRST_NAME = :vIme) OR (:vIme IS NULL)))`. Below the tree view, there are buttons for 'Add Item', 'Add Group', 'Add Criteria', 'Add Named Criteria...', and 'Delete'. The 'Criteria Item' section at the bottom of the dialog is configured as follows:

- Conjunction: AND
- Attribute: FirstName
- Operator: Equals
- Operand: Bind Variable
- Parameter: vIme
- Ignore Case:
- Ignore Null Values:
- Validation: Optional

At the bottom of the dialog, there are 'Help', 'OK', and 'Cancel' buttons.

Where uvjet ili View Criteria

View Criteria – filter koji se odnosi na instancu view objekta.

- af:query panel – forma za traženje
- filtriranje instance pri pokretanju u AM-u

Where uvjet – filter koji se postavlja u dizajnu view objekta, te se odnosi na sve njegove instance

AM = transakcija

The screenshot displays the Oracle ADF IDE interface. At the top, a window title bar shows several open files: 'g2011 Overview', 'Business Components Diagram.adfbc_diagram', 'DepartmentsEO.xml', 'DepartmentsVO.xml', 'EmployeesVO.xml', and 'Hroug2011AM.xml'. The main workspace is divided into two panes. The left pane, titled 'Data Model Components', contains a 'Data Model' section with instructions: 'Select a view object from the Available tree, select the desired parent instance or application module from the Data Model tree, and use the shuttle control to create a named view instance in the data model.' Below this is a 'View Object Instances' section with the text: 'The data model contains a list of view object and view link instances, displaying master-detail relationships.' The right pane, titled 'Data Model', shows a tree structure under the application module 'Hroug2011AM'. It contains three 'Departments' instances (Departments1, Departments2, Departments3) and three 'Employees' instances (Employees1, Employees2, Employees3). Dotted lines indicate master-detail relationships between the departments and employees. Below the panes are input fields for 'New View Instance:' and 'View Instance:'. The 'View Instance:' field has a small edit icon to its right.

Pushme-Pooling

Edit Configuration

Configuration Name: Hroug2011AMLocal

Application Module | **Pooling and Scalability** | Properties

Application Pool

Initial Pool Size	0
Maximum Pool Size	4096
Referenced Pool Size	10
Minimum Available Size	5
Maximum Available Size	25
Idle Instance Timeout (s)	600
Pool Polling Interval (s)	600

Connection Pool

Initial Pool Size	0
Maximum Pool Size	4096
Minimum Available Size	5
Maximum Available Size	25
Idle Instance Timeout (s)	600
Pool Polling Interval (s)	600

Failover Transaction State Upon Managed Release

Disconnect Application Module Upon Release

Support Dynamic JDBC Credentials

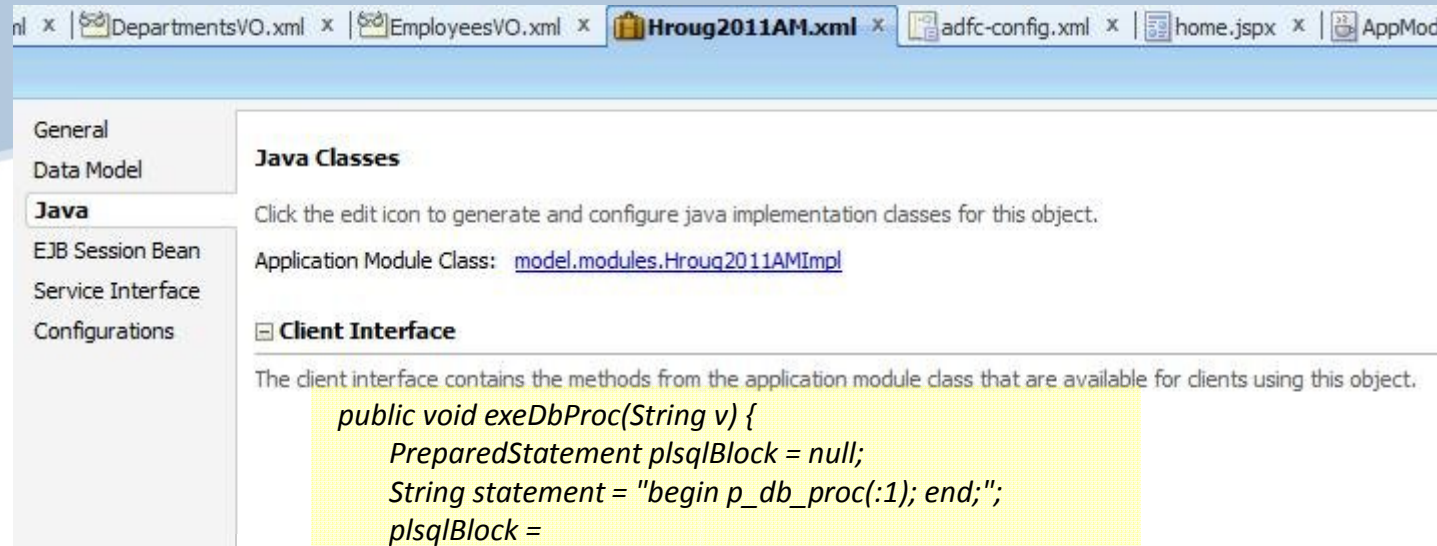
Reset Non-Transactional State Upon Unmanaged Release

Enable Application Module Pooling

Reset

Help OK Cancel

Drag'n'Drop Java



General
Data Model
Java
EJB Session Bean
Service Interface
Configurations

Java Classes

Click the edit icon to generate and configure java implementation classes for this object.

Application Module Class: [model.modules.Hroug2011AMImpl](#)

Client Interface

The client interface contains the methods from the application module class that are available for clients using this object.

```
public void exeDbProc(String v) {
    PreparedStatement plsqliBlock = null;
    String statement = "begin p_db_proc(:1); end;";
    plsqliBlock =
getDBTransaction().createPreparedStatement(statement,0);
    try
    {
        plsqliBlock.setString(1,v);
        plsqliBlock.execute();
    }
    catch (SQLException e) {
        throw new JboException(e);
    } finally {
        //if (st != null) {
        try {
            // 10. Close the JDBC CallableStatement
            plsqliBlock.close();
        }
        catch (SQLException e) {}
    }
}
```

Java na izložbi

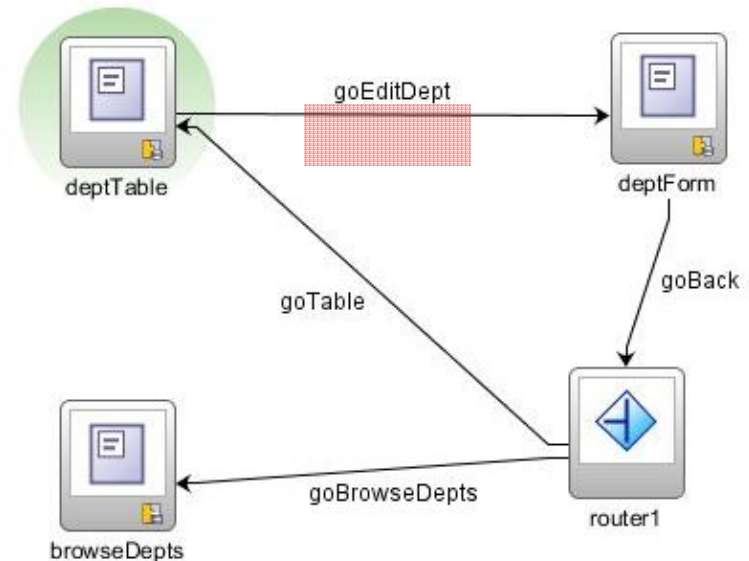
The screenshot shows the Oracle ADF IDE interface. At the top, several browser tabs are open, including 'Hroug2011AM.xml' and 'Hroug2011AMImpl.java'. The main workspace displays the 'Java Classes' view with a message: 'Click the edit icon to generate and configure java implementation classes for this object.' Below this, the 'Client Interface' view is partially visible. A dialog box titled 'Edit Client Interface' is open in the foreground. It contains two panes: 'Available:' and 'Selected:'. The 'Available:' pane lists 'exeDbProc(String):void'. The 'Selected:' pane lists 'Hroug2011AM'. Below the panes are fields for 'Parameter:', 'Type:', and 'Element Java Type:'. At the bottom of the dialog are 'Help' and 'OK' buttons. A red arrow labeled '1.' points to the edit icon (a pencil) in the top right corner of the 'Edit Client Interface' dialog. Another red arrow labeled '2.' points to the 'Data Controls' tree in the bottom right corner of the IDE. The 'Data Controls' tree shows a folder 'Hroug2011AMDataControl' containing several items: 'Departments 1', 'Employees 1', 'exeDbProc(String)', and 'Operations'.

Task Force...ups...Flow

Task Flows

- Prikazuju navigaciju između stranica
- Mogućnost pozivanja metoda (operacija)
- Mogućnost pozivanja ostalih task flow-a
- Upravljanje transakcijama
- Prosljeđivanje parametara
- Više bounded task flow-ova i jedan unbounded Task flow po aplikaciji

Bounded Task Flow



Task Flows are the key vehicle for creating encapsulated, stand alone services with a User Interface that can be consumed in multiple pages in various applications.

Lucas Jellema, Oracle ACE director

Page, fragment, jsp, jsff

Jsp pages

- Samostalni moduli
- Najlakše ih je napraviti i razumijeti
- Ne mogu se koristiti unutar regije, pa imaju malu iskoristivost.

JSF fragments

- Definicije stranica koje su implementirane unutar postojeće jsp stranice.
- Koriste se u kontekstu regija.

ADF regions

- Sastoje se od af:region taga, te Bounded task flowa i fragmenta

Page templates

- Page templates su definicija sučelja koji se koriste kako bi se osigurala konzistentnost aplikacije.

Drag'n'Drop Rock'n'Roll

The screenshot displays the Oracle ADF IDE interface. At the top, the title bar shows several open files: 'g2011 Overview', 'Business Components Diagram.adfbc_diagram', 'DepartmentsEO.xml', 'DepartmentsVO.xml', 'EmployeesVO.xml', and 'Hroug2011AM.xml'. The main workspace is divided into several panes:

- Data Model Components:** Contains instructions for selecting view objects and creating named view instances.
- Available View Objects:** Lists 'model.Model' and 'model.uiview'.
- Data Model:** Shows a tree structure for 'Hroug2011AM' with sub-nodes for 'Departments1', 'Employees3', 'Employees1', 'Departments2', and 'Employees2'.
- Application Resources:** Lists configuration files like 'adfc-config.xml', 'faces-config.xml', and 'trinidad-config.xml'.
- Data Controls:** Shows a tree for 'Hroug2011AMDataControl' with sub-nodes for 'Departments1', 'Employees1', 'exeDbProc(String)', and 'Operations'.
- Recently Opened Files:** Lists 'deptTable.jsff - Structure' and 'jsp:root'.

A context menu is open over the 'Table' option in the Data Controls pane. The menu items are:

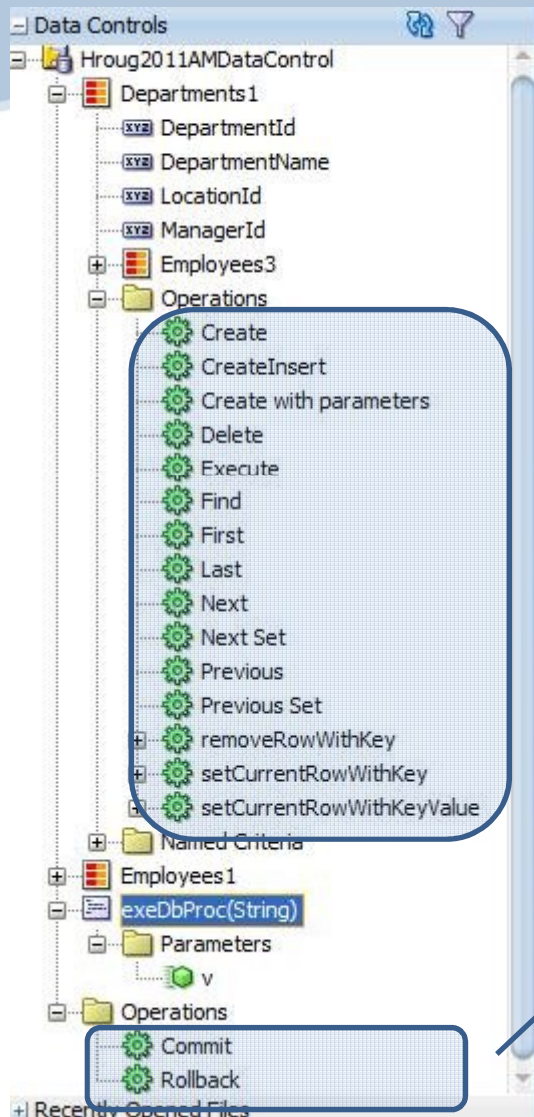
- Carousel
- Form
- Gantt
- Gauge...
- Geographic Map
- Graph...
- Hierarchy Viewer...
- Multiple Selection
- Navigation
- Single Selection
- Table** (highlighted)
 - ADF Table...
 - ADF Read-only Table...
 - ADF Read-Only Dynamic Table
 - ADF Pivot Table...
- Thematic Map...
- Tree
- Cancel

Two red arrows indicate the drag-and-drop operation: one points from the 'Table' option in the context menu to the 'Departments1' node in the Data Controls pane, and the other points from the 'Departments1' node to the 'Table' option in the context menu.

Drag'n'Drop Rock'n'Roll

The screenshot illustrates the process of adding a data control to an ADF form. On the left, the 'Data Controls' pane shows a project named 'Hroug2011AMDDataControl' with a data control 'exeDbProc(String)' under the 'Parameters' folder. A blue box highlights this data control, and a blue arrow points from it to a 'Create' context menu. The menu has 'Method' selected, and another blue box highlights the 'ADF Parameter Form...' option. A second blue arrow points from this menu option to a form in the center. The form contains input fields for 'DepartmentId', 'DepartmentName', 'ManagerId', and 'LocationId', each with a corresponding '# {...}.inputValue' binding. Below these are two buttons: 'Spremi' and 'Povratak'. A dashed box highlights the area below the buttons, and a blue box highlights a new data control being added to this area. The new data control has a binding '# {...}.v...label' and '# {...}.v.inputValue', and is labeled 'exeDbProc'.

Drag'n'Drop Rock'n'Roll



Operacije koje se odnose na pojedinu instancu View objecta

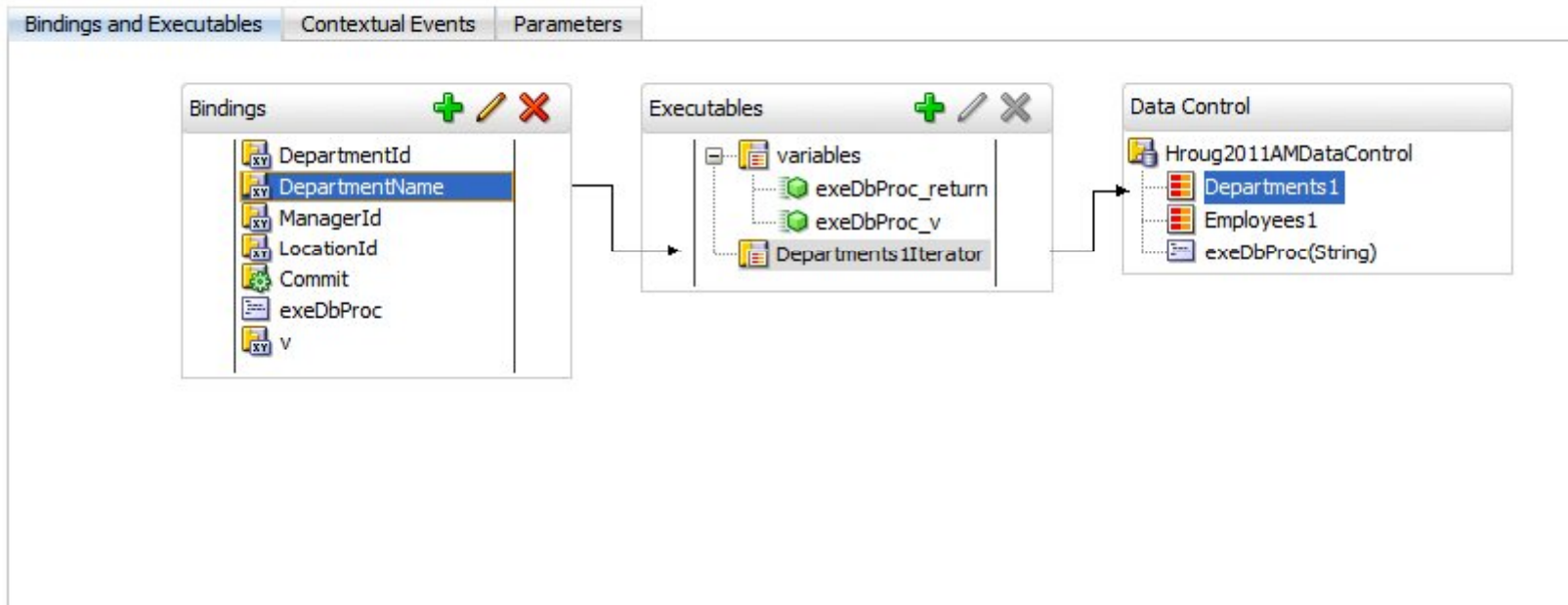
Operacije koje se odnose na Application modul, tj. Transakciju/konekciju

Page Definiton

Page Data Binding Definition

This shows the Oracle ADF data bindings defined for your page. Select a binding to see its relationship to the underlying Data Control.

Page Definition File: [fragments/deptFormPageDef.xml](#)



Pregled svih operacija, varijabli, data setova za pojedinu stranicu/fragment.

Bindings je pojam koji opisuje poveznost određene ui kontrole s modelom (podataka).

Lahko je navigat!

Departments1

menus View Edit Detach

DepartmentId	DepartmentName	ManagerId	LocationId
#{...DepartmentId}	#{...DepartmentNam	#{...ManagerId}	#{...LocationId}
#{...DepartmentId}	#{...DepartmentNam	#{...ManagerId}	#{...LocationId}
#{...DepartmentId}	#{...DepartmentNam	#{...ManagerId}	#{...LocationId}

Columns Hidden Columns Frozen statusbar

Employees3

menus View toolbar Detach

EmployeeId	FirstName	LastName	Email	PhoneNumber
#{...EmployeeId}	#{...FirstName}	#{...LastName}	#{...Email}	#{...PhoneNu
#{...EmployeeId}	#{...FirstName}	#{...LastName}	#{...Email}	#{...PhoneNu
#{...EmployeeId}	#{...FirstName}	#{...LastName}	#{...Email}	#{...PhoneNu

Toolbar Button - Edit - Property Inspector

Find

Common

- Id: ctb1
- Rendered: <default> (true)
- Type: <default> (default)
- Selected: <default> (false)
- Icon:
- Text: Edit

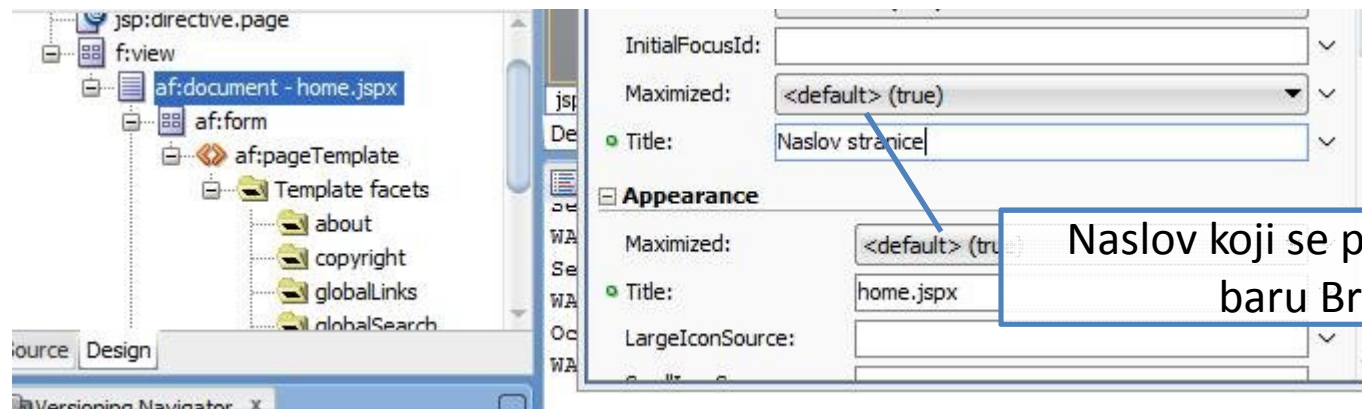
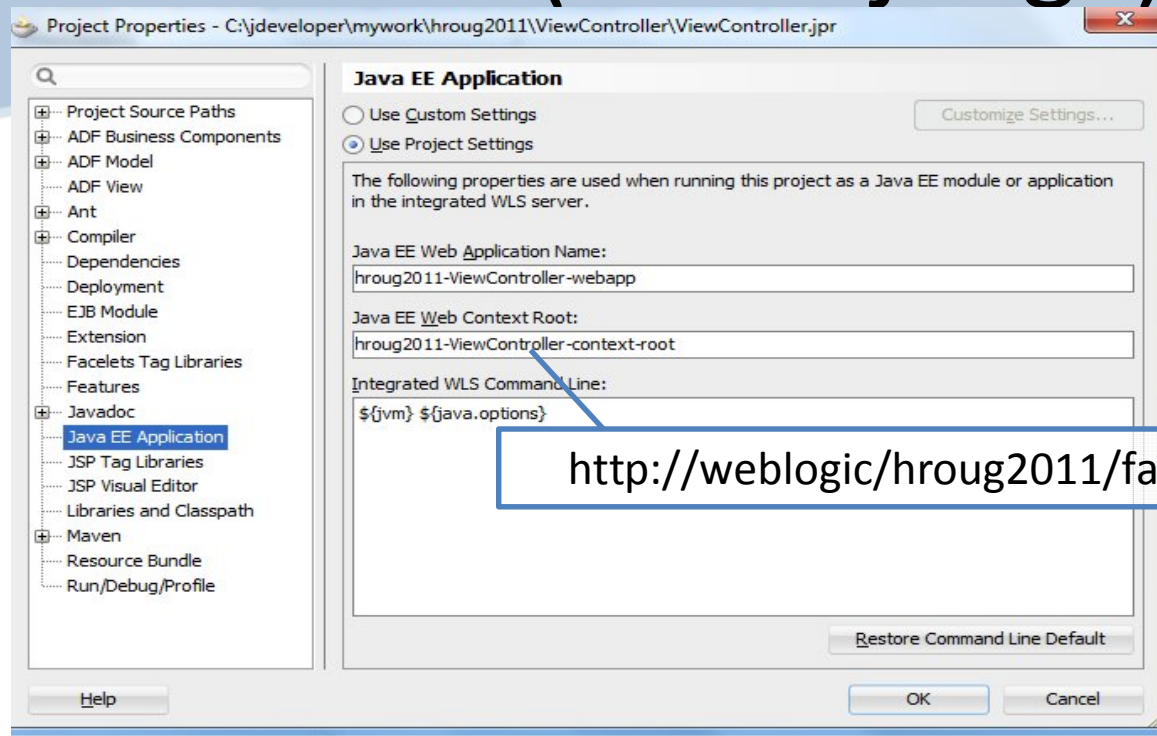
Button Action

- Action: goEditDept
- ActionListener:
- UseWindow: <default> (false)
- WindowEmbedStyle: <default> (window)
- WindowModalityType: <default> (modeless)
- WindowHeight:
- WindowWidth:

Appearance

- Type: <default> (default)

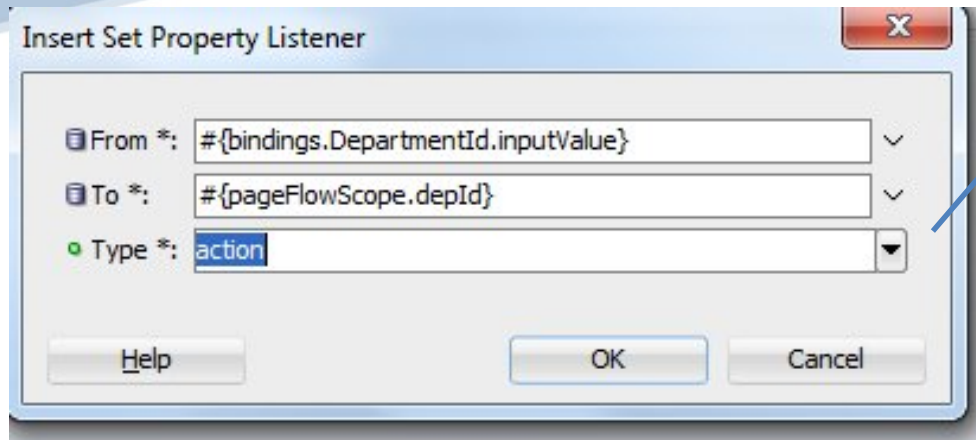
Tips&Tricks (imenujte ga)



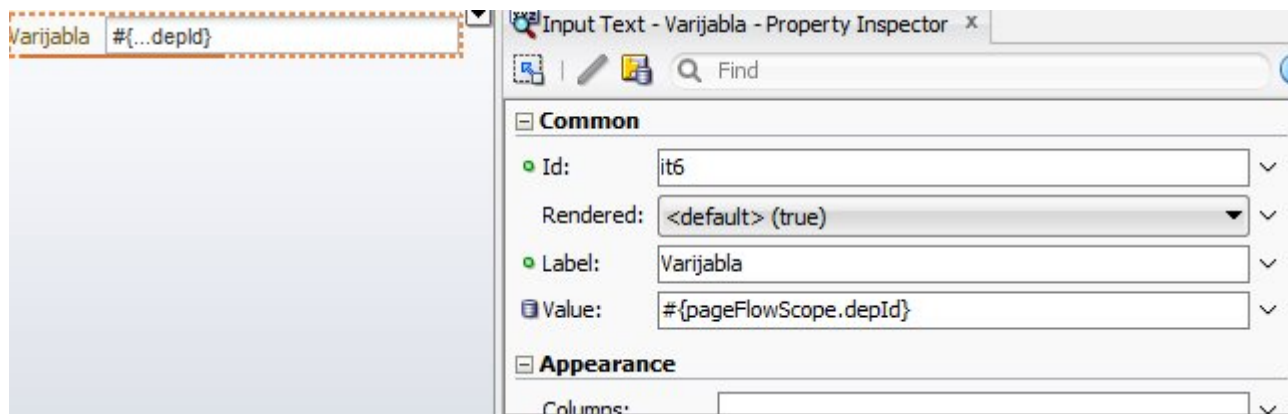
Tips&Tricks (ADF library)

The screenshot illustrates the process of creating a deployment profile for an ADF library. In the top-left, a context menu is open over a project, with the 'Deploy' option selected. This leads to the 'Create Deployment Profile' dialog box, which is shown in the top-right. The dialog is configured with 'ADF Library JAR File' as the profile type, 'adflib1801' as the name, and a description: 'Creates a profile for deploying ADF components as an Application JAR file. The resulting jar can then be reused in ADF applications, or to build other ADF Library jars. This is done without'. Below the dialog, a file explorer shows a project structure with a file named 'adflibHelper.jar' selected. A context menu is open over this file, with 'Add to Project...' highlighted. Two large red curved arrows indicate the flow from the 'Deploy' menu item to the dialog, and from the dialog to the 'Add to Project...' menu item.

Tips&Tricks (postavljanje varijabli)



Npr. Kod klika na Button.
Uzima vrijednost iz
Bindingsa i puni varijablu
depId u pageFlow scopu.



Tips&Tricks (Partial refresh)

The image shows two windows from an IDE. The left window is the 'Input Text - Varijabla - Property Inspector'. It has a search bar and several property sections: 'ReadOnly', 'Disabled', 'AutoSubmit', 'AutoComplete', 'AutoTab', 'PartialTriggers', 'Validation', and 'Contextual Events'. The 'PartialTriggers' field is currently empty. The right window is the 'Edit Property: PartialTriggers' dialog. It contains a text area with instructions: 'Shuttle items from Available to Selected to define the list of scoped ids for this component. If an item cannot be shuttled, it requires an Id or an ancestor naming container requires an Id. Note that scoped ids use a combination of relative and absolute path references depending on their relationship to the current component.' Below this are two lists: 'Available' and 'Selected'. The 'Available' list contains a tree view with items: 'root', 'panelFormLayout - pf1', 'panelFormLayout - pf2', 'inputText - it3', 'commandButton - cb1', and 'inputText - it6'. The 'Selected' list contains 'commandButton - cb1'. A red arrow points from the 'PartialTriggers' field in the Property Inspector to the 'Selected' list in the dialog. At the bottom of the dialog are 'Help', 'OK', and 'Cancel' buttons. The status bar at the bottom of the IDE shows 'Analyzer Instances' and 'Pending Changes (Subversion)'.

Zaključak

- Potpuno novi način razvoja aplikacija
- Don't upgrade, inovate!
- Prilagodba MVC-u
- Nova razvojna okolina
- Weblogic izazov

Iako dijeli principijelne sličnosti s Formsima, ADF pruža potpuno novi oblik razvoja aplikacija, i potrebno je uložiti mnogo truda za njegovo usvajanje.

Unatoč tome ADF je vrlo moćna tehnologija, i svako ulaganje u njeno usvajanje će se višestruko vratiti.

10 razloga zašto ADF još nije toliko raširen koliko bi Oracle volio

1. ***“The technology changes/improves too rapidly, and I have fears of 1) ‘will what I am doing become old-school soon’ and 2) ‘is a new/better way of doing going to be released within a few months’.”***
2. “The documentation is daunting.”
3. “The suggested pre-requisites and structured learning programs seem to change every few months.”
4. “I have spent considerable time and money hiring smart coders that don’t take to 4GL development as readily.”
5. “Code generating technologies have some negative connotations when it comes to code cleanliness, efficiency, and performance.”
6. ***“There are ‘too many’ choices. Just show me the right way and the best practice.”***
7. “Deeper customization required to fit my specific implementation needs are much less 4GL and have a steeper development learning curve.”
8. “There aren’t enough ‘experts’ readily available in the market that really know what they’re doing that would drive my ability to hire resources for low-to-medium cost, schedule a rapid delivery (ramp up, implement, deliver, support), or ensure what is being done is being done the ‘right way’.”
9. “ADF is an end-to-end framework, so I need resources that understand the entire stack to be on the same page about the approach we are taking.”
10. ***“This is nothing like anything I have worked with before.”***

