Oracle JHeadstart Forms2ADF - Overview
JHeadstart Forms2ADF Generator

• Generates ADF Business Components based on Forms Data Usages
  • ADF BC Entity Objects created for used tables
  • ADF BC View Objects created for blocks and record group queries
    • Query Bind parameters created based on references to :block.item
    • Model LOV’s created for each LOV / Record Group
  • ADF BC Application Module created for each form
• Extracts Forms User Interface Definitions into JHeadstart Application Definition
  • Groups created for each block
  • Group Items created for each item in a block
  • (Stacked) region containers and regions created based on item placement on (tabbed) canvasses and within framed graphics
  • Domains created based on forms item allowable values
  • PL/SQL logic copied as “documentation” nodes
Forms2ADF Generation Process

Application Definition

ADF Business Components
ADF Model
Data Controls
Data Bindings
JSF + ADFc
JSF JSP ADF Faces

View
Controller
Model
Business Services

Form (.fmb)

JHeadstart Forms2ADDF Generator
Forms2ADDF Generation Process

- Application Definition
- Generator Templates
- JHeadstart Application Generator
- JSF JSP
  - ADF Faces
- JSF+ADFc
- ADF Model
  - Data Bindings
  - Data Controls
- ADF Business Components

View
Controller
Model
Business Services
Running the Forms2ADF Generator

The New Gallery window is open, showing a list of technologies and categories. The selected technology is "JHeadstart Forms2ADF Generator," which launches the JHeadstart Forms2ADF generator wizard. This wizard allows you to generate ADF Business Components and a JHeadstart Application Definition from Oracle Forms definitions (elements and properties).

To enable this option, you must select a project in the Application Navigator.
Running the Forms2ADF Generator

Welcome to the JHeadstart Forms2ADF Generator Wizard

This wizard will help you to generate ADF Business Components based on the forms you select. In addition, a JHeadstart Application Definition File will be generated that you can use to generate a fully functional ADF application.
Running the Forms2ADF Generator

Select the form modules (.fmb or .xml format created with FRMF2XML utility with command line argument 'USE_PROPERTY_IDS=yes') for which you want to generate ADF Business Components and JHeadstart metadata.
Running the Forms2ADF Generator

Exclude elements from the selected forms

Enter elements from the selected forms to be excluded when generating ADF Business Components and the JHeadstart Application Definition File. You can use the * at the end of the name to match elements that start with the excluded element name.

Names of form elements to be ignored by Forms2ADF Generator:

<table>
<thead>
<tr>
<th>Name</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALENDAR</td>
<td></td>
</tr>
<tr>
<td>CALENDAR_TIME</td>
<td></td>
</tr>
<tr>
<td>QMS$*</td>
<td></td>
</tr>
<tr>
<td>OF_*</td>
<td></td>
</tr>
<tr>
<td>CG$CTRL</td>
<td></td>
</tr>
<tr>
<td>ORAPORTMSFACES*</td>
<td></td>
</tr>
<tr>
<td>TOOLBAR*</td>
<td></td>
</tr>
</tbody>
</table>
Running the Forms2ADF Generator

ADF Business Components Database Connection

Select the connection to your database for ADF Business Components. This connection should have access to all tables and views used in the forms you selected.

If the connection uses synonyms to access tables or views used by the forms, then the connection user must have the SELECT ANY DICTIONARY privilege for the Forms2ADF generator to succeed.

Connection: ocm

User Name: ocm

Driver: oracle.jdbc.OracleDriver

Connect String:jdbc:oracle:thin:@localhost:1521:orcl
Running the Forms2ADF Generator

ADF Business Components Settings:
- Entity Objects Package: model.entities
- Entity Associations Package: model.entities.assoc
- View Objects Package: model.queries
- View Links Package: model.queries.lnk
- Lookup View Objects Package: model.queries.blok
- Application Modules Package: model.service
- Root Application Module: OcmdenoService
Running the Forms2ADF Generator

JHeadstart Settings

Service Name: Ocmdemo
View Controller Project: ViewController.jpr
Extract PL/SQL Logic?: Yes
Migrate Text Graphics?: Yes
Use Text Graphics as Label?: Yes
LOV implementation: ADF Model
Search implementation: ADF Model
Running the Forms2ADF Generator

You have finished setting up the Forms2ADF generator.

- ADF Business Components database connection: ocm
- Entity Objects Package: model.entities
- View Objects Package: model.queries
- Application Modules Package: model.service

Selected Form modules:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>file:/J:/ocm/Forms/OCM010.fnb</td>
<td>Form</td>
</tr>
<tr>
<td>file:/J:/ocm/Forms/OCM050.fnb</td>
<td>Form</td>
</tr>
<tr>
<td>file:/J:/ocm/Forms/OCM0110.fnb</td>
<td>Form</td>
</tr>
<tr>
<td>file:/J:/ocm/Forms/OCM0120.fnb</td>
<td>Form</td>
</tr>
</tbody>
</table>

Click Finish to generate the ADF Business Components and JHeadstart metadata, or alternatively click Back to review and change the options.
Generated ADF Business Components
Generated Application Definition
## Administrate People

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ken</td>
<td>Atkins</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frank</td>
<td>Brink</td>
<td><a href="mailto:frank.brink@oracle.com">frank.brink@oracle.com</a></td>
<td></td>
</tr>
<tr>
<td>Bradley</td>
<td>Brown</td>
<td><a href="mailto:david.brown@oracle.com">david.brown@oracle.com</a></td>
<td></td>
</tr>
<tr>
<td>Steven</td>
<td>Davolael</td>
<td><a href="mailto:stevan.davolael@oracle.com">stevan.davolael@oracle.com</a></td>
<td></td>
</tr>
<tr>
<td>Paul</td>
<td>Dorsev</td>
<td></td>
<td></td>
</tr>
<tr>
<td>William</td>
<td>Dwight</td>
<td><a href="mailto:william.dwight@oracle.com">william.dwight@oracle.com</a></td>
<td></td>
</tr>
<tr>
<td>Kent</td>
<td>Gradano</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Erwin</td>
<td>Groenenda</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Martijn</td>
<td>Hinzen</td>
<td><a href="mailto:martijn@cumquat.nl">martijn@cumquat.nl</a></td>
<td></td>
</tr>
</tbody>
</table>

### ODTUG History

- Country: Netherlands
- Birthdate: 21-Sep-1963
- Biography: He created Headstart, he created JHeadstart and he will create much more.
Oracle Forms Screen

Title: CDM RuleFrame: the framework for analysing, designing and implementing business rules
Technology Track: Oracle Forms Developer
Keywords: business rules, design, J2ee, HTML, forms, eee
Presentation Level: Novice

Abstract: Business Rules are key. Multiple technologies (Java/J2ee/HTML, Forms, Portal) used to build applications that act on the same data.

Schedule
Room: Capri
Slot: 2:00 - 3:00 PM
Day: Wednesday, 19th June

Presenters
First Name: Steven
Last Name: Davolear
Primary presenter?

First Name: Lucas
Last Name: Jellema
Primary presenter?

Record: 1/2
JHeadstart Generated ADF/JSF Page

Presentation:
Title: COM RuleFrames: the framework for analysing, designing and implementing business rules
Technology Track: IS
Keywords: business rules, design, J2EE, HTML, forms, XML
Presentation Level: Novice
Abstract: Business Rules are key. Multiple technologies (J2EE/HTML, Forms, Portal) used to build applications that act on the same data.

Schedule:
Room: Capri
Slot: 2:00 - 3:00 PM
Day: Wednesday 10th June

Presenters:
First Name | Last Name | Primary Presenter? | Demonstrator?
--- | --- | --- | ---
Lucas | Jelena | ✓ | ✓
Steven | Developer | |
Oracle Forms Screen

ODTUG Conference Manager, 2002 Demo Application

Title: CDM RuleFrame: the framework for analysing, designing and implementing business rules
Technology Track: Oracle Forms Developer
Keywords: business rules, design, J2ee, html, forms, eee
Abstract: Business Rules are key. Multiple technologies (Java/J2ee/HTML Forms) result in applications that act on the same data.

Schedule
Room: Roman Ballroom Salon II
Slot: 2:00 - 3:00 PM
Day: Wednesday 19th June

Administraton

Preseneters
First Name: Steven
Last Name: Davelaar
Primary Presenter: Yes

Find %

Conference Rooms

OK Cancel

Name
Roman Ballroom Salon I
Florentine Ballroom
Pompeian Ballroom Salon II
Anzio
Turin
Trevi
Caori

Find

ORACLE
## Presentation Scheduler

<table>
<thead>
<tr>
<th>Title</th>
<th>Slot</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Ruleframe: the framework for analyzing, designing</td>
<td>2.00 - 3.00 PM</td>
<td>Wednesday 19th June</td>
</tr>
<tr>
<td>Forms Tuning Techniques -- Users Will Sing Praise</td>
<td>2.00 - 3.00 PM</td>
<td>Wednesday 19th June</td>
</tr>
<tr>
<td>High Performance Web Sites with Web Cache</td>
<td>3.15 - 4.15 PM</td>
<td>Tuesday 18th June</td>
</tr>
<tr>
<td>Implementing Business Rules in Business Components</td>
<td>4.45 - 5.15 PM</td>
<td>Wednesday 19th June</td>
</tr>
<tr>
<td>Java Race: Case Study Comparing the Performance of Java/3EE</td>
<td>8.30 - 9.30 AM</td>
<td>Monday 17th June</td>
</tr>
<tr>
<td>Java/3EE Application Development with Oracle Design</td>
<td>11.00 - 12.30 AM</td>
<td>Wednesday 19th June</td>
</tr>
<tr>
<td>Murate to the Web with Oracle 9 Designer</td>
<td>3.15 - 4.15 PM</td>
<td>Monday 17th June</td>
</tr>
<tr>
<td>PM 1: I am in Vegas - Business Intelligence for the millenium</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Details

**Abstract:**
For developing batch processing applications, both PL/SQL and Java can be used. For both languages, a number of technical setups is possible, including native compilation. This...

**Keywords:**
java, pl/sql, performance, batch

**OraclePresentation:**
- Yes

### Audience

- [ ]

### Presenters

- [ ]

### Ambassadors

- [ ]
What about PL/SQL Logic?

- Custom PL/SQL Logic not automatically migrated to Java
- JHeadstart prefers sound multi-tier target architecture (ADF MVC) above automated PL/SQL migration
  - Analyze the logic
  - Determine in which tier(s) it should be implemented
  - Determine how to implement it
- Application Definition Editor shows all form, block and item triggers and all program units
  - Good overview of all custom PL/SQL logic
  - Ability to move logic to database, to ADF BC or to JSF managed beans
PL/SQL Code in Appl. Definition Editor

/* CGFK$QRY/people_psn_cpy_fk */
/* Query foreign key value/query lookup data. */
PROCEDURE CGFK$QRY_PEOPLE_PSN_CPY_FK(
    p_l_cpy_name varchar2, /* Value in item:PEOPLE.L_CPY*/
    p_id_works_for in number, /* Value in item:PEOPLE.ID_WORK*/
)
IS
CURSOR C1 IS
    SELECT L_CPY.NAME
    FROM OCM_COMPANIES L_CPY
    WHERE P_ID_WORKS_FOR = L_CPY.ID
;
CURSOR C2 IS
    SELECT L_CPY.NAME
    FROM OCM_COMPANIES L_CPY
    WHERE P_ID_WORKS_FOR = L_CPY.ID
;
BEGIN
    /* */
END;
JHeadstart
OraFormsFaces
Generator
OraFormsFaces

- Third party product supplied by Commit Consulting
- Allows reuse of existing Forms as full featured JSF components
- Two-way communication between forms and ADF Faces web pages
- Allows for *incremental migration* to ADF/SOA world
JHeadstart OraFormsFaces Generator

- JHeadstart accelerates use of OraFormsFaces
  - Generate pages with embedded Form components using Item display type “OraFormsFaces”
  - Easy deeplinking into Oracle Forms
  - Generate Create, Delete, Save buttons that call out to embedded Oracle Form
  - Add security to JSF pages holding Forms Components
  - Generate menu structure
Running the OraFormsFaces Generator

The OraFormsFaces Generator is a tool that allows you to generate ADF Faces pages which embed Oracle Forms using the OraFormsFaces technology. To enable this option, you must select a project in the Application Navigator.
Running the OraFormsFaces Generator

Select the form modules (.fmb or .xml format created with FRMF2XML utility with command line argument ‘USEPROPERTY_IDS=yes’).

Selected Form modules:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>file:/J:/ocm/forms/OCM0020.fmb</td>
<td>Form</td>
<td></td>
</tr>
<tr>
<td>file:/J:/ocm/forms/OCM0030.fmb</td>
<td>Form</td>
<td></td>
</tr>
<tr>
<td>file:/J:/ocm/forms/OCM0040.fmb</td>
<td>Form</td>
<td></td>
</tr>
<tr>
<td>file:/J:/ocm/forms/OCM0130.fmb</td>
<td>Form</td>
<td></td>
</tr>
<tr>
<td>file:/J:/ocm/forms/OCM0140.fmb</td>
<td>Form</td>
<td></td>
</tr>
</tbody>
</table>

Browse

< Back  Next >  Finish  Cancel
Running the OraFormsFaces Generator
OraFormsFaces Metadata in JHeadstart Application Definition
Integration JHeadstart - OraFormsFaces
Setting up Deeplinking between ADF Page and Oracle Form

![Diagram showing ADF Page and Oracle Form configuration](image)

- **General**
  - Bound to Model Attribute: * ✓
  - Name: * LastName
  - Attribute Name: LastName
  - Value:
    - Java Type: java.lang.String
  - Display Type: * groupLink
  - Link Group Name: jdbocm.OCM0140

- **Display Settings**
  - Display in Form Layout: * False
  - Display in Table Layout: * True
Setting up Deeplinking between ADF Page and Oracle Form
Setting up Deeplinking between ADF Page and Oracle Form

![Diagram showing ADF Page and Oracle Form integration with DeepLinking](Image)
Integration JHeadstart – OraFormsFaces Deeplinking JSF Page and Oracle Form
Integration JHeadstart – OraFormsFaces
Deeplinking JSF Page and Oracle Form
JHeadstart – Key Messages

• JHeadstart incorporates a host of best practices in developing ADF Fusion web applications
• Rapid prototyping to speed up analysis phase, and to figure out most suitable user interface patterns
• Repetitive development tasks are automated.
• JHeadstart auto-implements common user interface design patterns
• Forms application definitions can be reused to ease transition to ADF and SOA
• Learn ADF 11 by example.