Oracle Developer Day

Session 4
Building the Business Logic

Speaker
Speaker Title
Agenda

- Building the Business Model
- Focussing on Entity Objects
- Building View Objects
- Assembling the Application Module
- Summary

Focus on ADF Business Components

An ideal entry point to web application development in the J2EE world…
ADF Architecture Overview

- Rich Clients
- Web and Wireless
- Controller
- Model (JSR 227)
- Business Services

ADF Business Components

- Complete business tier development framework
- Implements J2EE design patterns
- Best practice solutions for:
  - Business logic implementation
  - Scalability and performance
  - Optimized database interaction
Focussing on Entity Objects

Build the Model

- Build Entity Objects using Wizards, diagram and dialogs
- Set validation and display properties
Reviewing Familiar Concepts

- Format Masks
- Default Values
- Declarative Range Validation
- Validation from Code
- Transactional Triggers
- Visual Attributes
- List of Values

Add Display Details
Static Default Values

Dynamic Default Values
Dynamic Default Values - In the Java Class...

```java
/**
 * Add attribute defaulting logic in this method.
 */
protected void create(AttributeList attributeList) {
    super.create(attributeList);
    setHiredate(Dates.truncSysdate());
}
```

Comparison Validation
List Validation

Forms Item-Level Validation
Forms Record-Level Validation

Implement Custom Logic
Attribute-Level Validation Method
In Java Class

```java
public boolean validateThatNewDateIsNotInThePast(Date value) {
    if (value != null) {
        if (getPostState() == STATUS_NEW &&
            value.compareTo(Dates.truncSysdate()) <= 0) {
                return true;
            }
    }
    return false;
}
```

- Method Named `validateSomethingDescriptive`
- Takes a Parameter of the Correct Type for the Attribute
- Return boolean `true` if valid, and `false` if invalid
Validation Method In Java Class

```java
public boolean validateThatSNamesDontEarnOver4000() {
    if (getSal().intValue() > 4000 &&
        getEname().toUpperCase().startsWith("S")) {
        return false;
    }
    return true;
}
```

- Method Named `validateSomethingDescriptive`
- Return boolean true if valid, and false if invalid

Adding Validation Method
Considering Forms
Transactional Triggers

- PRE-INSERT, ON-INSERT, POST-INSERT
- PRE-UPDATE, ON-UPDATE, POST-UPDATE
- PRE-DELETE, ON-DELETE, POST-DELETE

doDML() Method In Java Class

```java
// Overrides default doDML() Behavior of Entity Object
protected void doDML(int operation, TransactionEvent e) {
    if (operation == DML_INSERT) {
        /* pre */
        super.doDML(operation, e); /* on */
        /* post */
    }
    if (operation == DML_UPDATE) {
        /* pre */
        super.doDML(operation, e); /* on */
        /* post */
    }
    // do the same for updates
}
```
Building View Objects

View Objects ~ Blocks

- Forms: Blocks have base table and non-base table or lookup table usages
- ADF: View objects query one or more tables
The Query

SQL Statement

By default, the SELECT list and FROM clause are automatically maintained. To override this mechanism, select Expert Mode.

Generated Statement

```sql
SELECT OrderItems.ORDERS_ID,
      OrderItems.LINE_ITEM_ID,
      OrderItems.PRODUCT_ID,
      OrderItems.UNIT_PRICE,
      OrderItems.QUANTITY,
      ProductInformation.PRODUCT_ID AS PRODUCT_ID,
      ProductInformation.PRODUCT_NAME
FROM Order_Items OrderItems, ProductInformation
WHERE OrderItems.PRODUCT_ID = ProductInformation.PRODUCT_ID
```

Query Classes

OrderItems. PRODUCT_ID = ProductInformation.PRODUCT_ID

Assembling the Application Module
Application Module ~ Form

- Maps to the Designer Module or Form
- Consists of one or many View Objects
- Create, using a wizard, dialog or a diagram

Reviewing the Structure

Data Model
Select a view object from the tree of available view objects, select the instance or application module to be its parent in the data model tree, and click "OK" to create a named instance of the view object in the data model.
Which Services to Expose?

Data Model

Select a view object from the tree of available view objects, select the instance or application module to be its parent in the data model tree, and click > to create a named instance of the view object in the data model.

Available View Objects:
- Model
- model.MP
- model.ED
- model.VO.reading
  - Employment
  - Department
  - Barcodes
  - Locations
  - Departments
  - Employee

Data Control Palette
- BusinessServices
  - Employee
  - Department
  - Locations

Data Binding

- Data Control Palette
  - Exposes business services
  - Available to any client
ADF Model

- Abstract service implementation from clients
- Declarative Data Binding
  - Common binding of data to UI components
  - Uses EL to reduce learning curve
  - Separates of view and model
- Declarative Data Validation
  - Consistent interface for common validation
  - Extendable for complex application validation
Summary

- ADF Business components provide many equivalent features familiar to Forms and Designer developers
- Develop business services
  - Declaratively through dialogs
  - Visually through diagrams
  - Wizard driven

Learn Oracle From Oracle

- Instructor led training
- Self-Study
- Online learning
- Oracle Certification
- Oracle iLearning
- Oracle Tutor

oracle.com/education