Oracle Developer Day

Session 3
Familiar Techniques: Modeling and Frameworks

Speaker
Speaker Title
Agenda

- Forms as a Framework
- Mapping Forms to Oracle ADF
- Familiar Concepts
- Phases of Application Development
- Full Development Life Cycle
- Development styles

You know Frameworks!

The task of any framework:
- Database interaction
- Business logic
- Application logic (flow)
- UI Logic
- Validation
- Security
- Data transfer and connectivity
- Data caching
- Transaction management
Forms Mapping to ADF

Java UI Renderer

Block UI Definition

Navigation Logic

Block Data Definition

Record Manager

SQL Interface

Rich Clients

Web, Wireless Clients

Controller

Model

Business Services

Forms Mapping to ADF

Java UI Renderer

Block UI Definition

Navigation Logic

Block Data Definition

Record Manager

SQL Interface

Rich Clients

Web, Wireless Clients

Controller

Model

Business Services
Using Familiar Concepts

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

Setting a Format Mask
Setting a Format Mask

Static Default Values
Static Default Values

How Do You Approach Development?
Forms and Designer: A Development Approach

1. Analyze and model business and data requirements
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow

Map the Approach to JDeveloper

1. Analyze and model business and data requirements
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow
Class Model

Sequence Model
Map the Approach to JDeveloper

1. Analyze and model business and data requirements
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow

Database Modeling and Generation

- Visualize tables, views on a diagram:
  - Relationships between tables
  - Column specification
  - Constraints
- Design and generate
- Import
- Reconciliation
  - CREATE statements
  - ALTER statements
- Browse offline schema
Map the Approach to JDeveloper

1. Analyze and model business and data requirements [Done]
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow

Forms as a Framework
Build Data Objects

- Reverse Engineered from DB
  - Wizard
  - UML Model
- New Object Model
  - Class model
  - Mapping
- Generated from Designer Repository

“Blocks and Forms”
Map the Approach to JDeveloper

1. Analyze and model business and data requirements [Done]
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow

Declarative Data Validation

- Add forms-like validation
  - Format masks
  - Client-side validation
- Extensible for complex application validation
Map the Approach to JDeveloper

1. Analyze and model business and data requirements [Done]
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow
Map the Approach to JDeveloper

1. Analyze and model business and data requirements [Done]
2. Define and model database layer
3. Build data objects based on data model (Blocks)
4. Add validation
5. Bind UI to data objects and refine UI
6. Build the Flow

Build Application Flow
Other Expectations?

Full Development Lifecycle

- Source Control
- UML Modeling
- Coding
- Testing
- Debugging
- Tuning
- Deployment
Team Development

- Create and apply patch
- Create local CVS repository
- Copy CVS root
- Refactoring Integration
- File extension registry
- Tag support
- Auto add files on commit
- Pending checkout window
- Compare viewer
- History viewer
- Global ignore list
- Remember check-in comments
- Integral CVS client
- Quick check-out
- CVS status dialog
- Separate CVS navigator

Visual and Declarative

- WYSIWYG Editors
  - JSF, JSP, HTML, Swing
- Visual Page Flow Editor
  - JSF, Struts
- Modelers
  - Visual DB, EJB, Class modeling
- Visual Editors
  - XML Schema, XSLT, WSDL
- Process Flow
  - BPEL Editor
Choice of Development Styles

Visual

Dialog

Code

Additional Capabilities

- Data and UI Split (Multiple Views)
- Coding Features
  - Code Completion, Code Insight, Code Templates
- Unit Testing
- One click deployment
- Tuning, auditing and profiling
- Extensible IDE
Summary

- As part of the Oracle Application Development Framework, ADF Business Components offers the Forms developer a familiar development environment
- ADF allows developers to work in the style of their choice
- JDeveloper offers developers using familiar development approaches
DEMONSTRATION

Join Over 3,000,000 Developers!
Free Technical Advice
Free Software Downloads

www.oracle.com/technology/products/jdev
otn.oracle.com/formsdesignerj2ee
Learn Oracle From Oracle

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

oracle.com/education

QUESTIONS ANSWERS