



November 1, 2002

Washington, DC



Understanding and Using BC4J Architecture in JDeveloper 9*i*

Dr. Paul Dorsey
Dulcian, Inc.
www.dulcian.com

Agenda

- ◆ Survey
- ◆ Overview
- ◆ Logical Structure
- ◆ Physical Structure
- ◆ Using UML
- ◆ Building with BC4J
- ◆ Development Strategies



Survey

- ◆ Who is using Oracle JDeveloper 9i?
 - in Test
 - in Production
- ◆ Who is using Java now?
 - 1-2 years
 - 2-5 years
 - 5-15 years
- ◆ Who has a need to move to Java now?





Overview

JDeveloper History

- ◆ Oracle bought some code for Borland's J-Builder in 1997.
- ◆ JDeveloper 1.0 (1998)
 - More or less useless
- ◆ JDeveloper 2.0 (1999)
 - Still useless



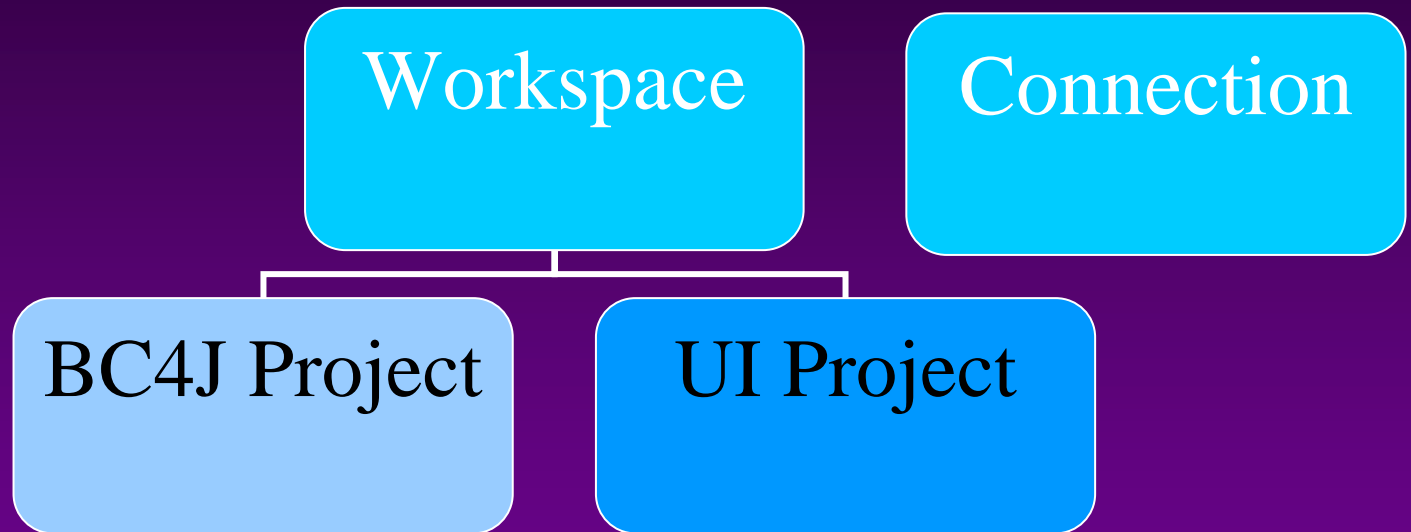
- ◆ JDeveloper 3.0 (2000)
 - BC4J & DAC
 - Finally the right idea!
- ◆ JDeveloper 3.2 (2001)
 - Bug fixes
 - First useful version
- ◆ JDeveloper 9i (2002)
 - First good version
 - Big rewrite
 - Change in direction

JDeveloper – What is it?

- ◆ Primarily a Java development environment
- ◆ Limited JSP support
- ◆ Main differentiation point – Business Components for Java (BC4J)



JDeveloper – Java Application



JDeveloper Application Architecture

◆ Workspace

➤ BC4J project

- Entities and associations
- Views and view links
- Module

➤ UI Project

- Class .java
 - Items
- Client data model .cpx

uses

Based on

◆ Connection

Projects

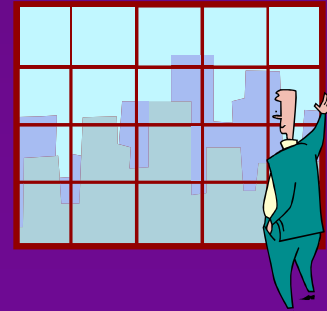
- ◆ Logical grouping of ... stuff
- ◆ Physically - a file with pointers (.jpr)
- ◆ Lives in one package (directory)
- ◆ Usually at least 2 in an application:
 - BC4J
 - UI or JSP





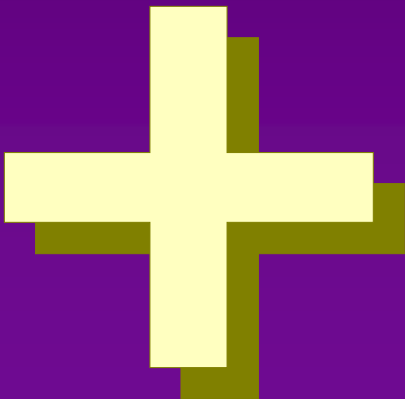
BC4J - Overview

- ◆ Main advantage over other development tools
- ◆ Primary way of handling database DML operations
- ◆ Overcomes hurdle of complex database interface to handle web-based applications
- ◆ BC4J features:
 - Builds a set of Java classes that wrap the DML commands and reference an XML document that stores the data structure
 - Classes make JDBC calls to the database to take over insert, update, delete and lock functions.



BC4J - Advantages

- ◆ BC4J classes can implement complex validation
 - Need to think carefully about whether this is the appropriate place for this function
- ◆ Reduced development time
- ◆ Can cache data to be shared among multiple users
 - Allows for tuning capabilities



What is BC4J?

◆ Business Components for Java (BC4J)

- Layer of business logic with reusable components
 - Main advantage over other Java development tools
 - Heart of the JDeveloper product
- Supported by programming protocol to interact with a relational database
- Primary way to handle DML operations
 - Inserts, Updates, Deletes, Queries
- Provides functional interface to database
 - Constructed using JDeveloper built-in code generators and wizards

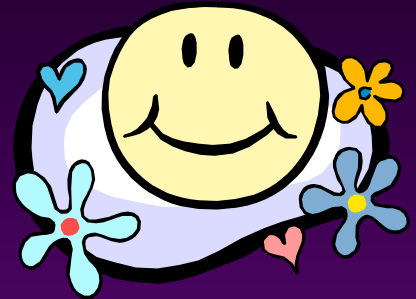
Before BC4J...

- ◆ Connecting Java-based applications to a relational database was very difficult
 - Complex JDBC code needed
 - Complex SQLJ needed
 - Need to maintain data caching
 - Batch updates
 - Track table locks and commits



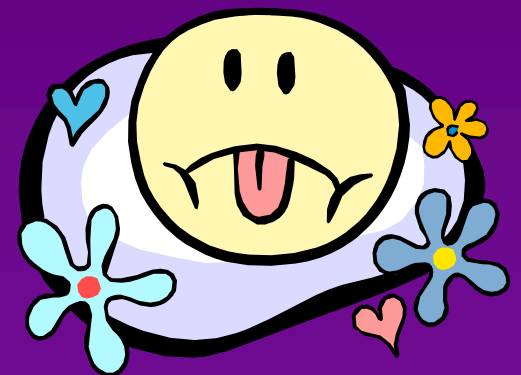
Advantages of Using BC4J Logic

- ◆ Maintains data cache in memory
 - Reduces number of database trips
 - Improves performance and scalability
- ◆ Allows you to write business logic in Java
- ◆ Database only handles data
 - Increase application modularity
- ◆ Tier-independent
 - Can be deployed as local Java Classes or Enterprise JavaBean (EJB) with no code rewriting
- ◆ Automatically implements J2EE BluePrints design patterns from Sun Microsystems.



Disadvantages of Using BC4J Logic

- ◆ Only enforced in applications using the BC4J layer
- ◆ Business logic in database is more robust.
- ◆ Need to weigh tradeoff between robustness and performance

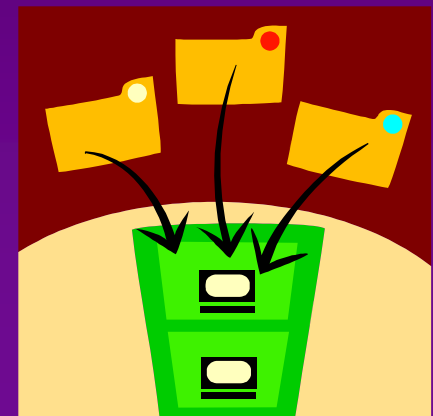




Logical Architecture

Three areas of BC4J

- ◆ 1. Entities and Associations
 - Cached copy of a portion of the database
 - An entity is linked to, at most, 1 table
- ◆ 2. View objects and view link objects
 - View objects that sit on top of entities (or nothing)
- ◆ 3. Data model
 - Collection of views
 - What the UI applications interact with



BC4J Structure

- ◆ Developers only need to think in terms of logical objects
- ◆ Manipulate Java classes only where complex validation or coding not handled by the wizards is required
- ◆ Conceptually divide application into 2 parts:
 - User interface and client logic
 - Database interface and business logic



Default Generator

DBMS Tables



Entities & Associations



Data Model

Dept

Emp 1 filtered view usage

Emp 2 unfiltered view usage

Views & view links



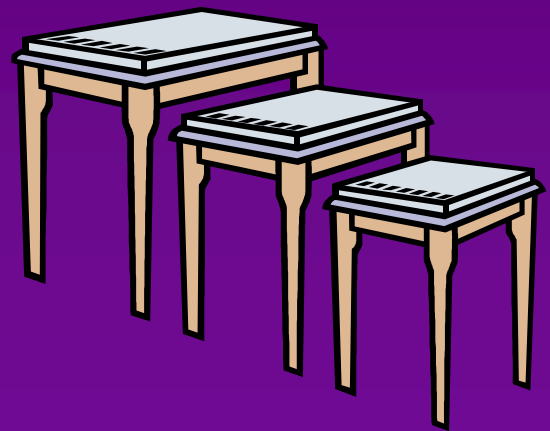


Entity Objects (1)

- ◆ Represent database tables, views, snapshots, synonyms
- ◆ Contain attribute (column) definitions
- ◆ Handle all business rules for tables and views
- ◆ Can be generated from existing objects or manually defined
- ◆ As of JDeveloper 9.03 – can also use EJBs and EJB Entity facades
 - EJBs – handle data persistence
 - Entity facades – handle business logic and client binding

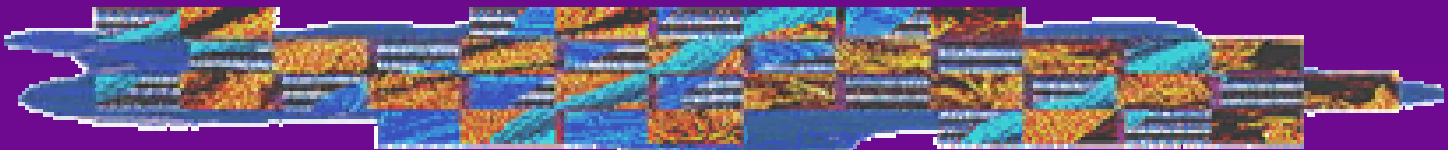
Entity Objects (2)

- ◆ Can differ from underlying database tables:
 - Do not need to represent all columns in a table
 - May have columns that are not in the database
 - May perform additional validation
 - May be defined to behave like views with INSTEAD-OF triggers
 - Need not be associated with any table



Attributes

- ◆ Properties of an entity object
- ◆ Property types are Java classes corresponding to column SQL types
- ◆ Two types:
 - Entity attributes – part of entity object corresponding to a table column
 - May include validation rules and business logic
 - View attributes – correspond to columns of the view object query result



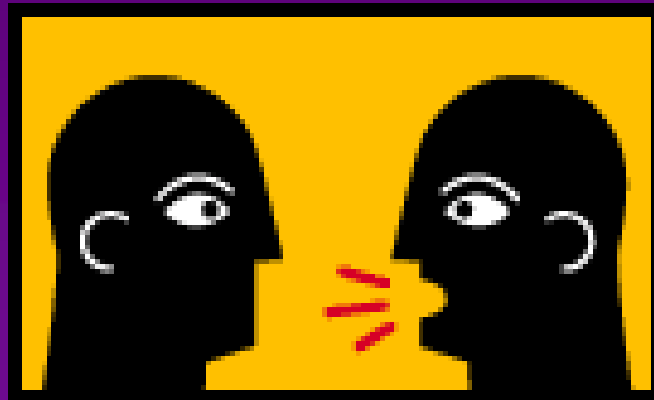
Associations (1)

- ◆ Define the relationship between pairs of entity objects
- ◆ Often represent foreign key constraints
 - Match “source” entity objects with “destination” entity objects
- ◆ Bi-directional
 - Can specify more than one attribute on each side of the association



Associations (2)

- ◆ Can differ from database referential integrity constraints
- ◆ May have a different cardinality from that in the database



BC4J Layer of Abstraction

◆ Limitations

- Entity object cannot directly point to more than one table or view
- Code in BC4J layer rather than database may be bypassed
- Commits development to product architecture

◆ Advantages

- Performance improvements
- Support for overloaded tables



View Objects

- ◆ Store SQL queries
- ◆ Use SQL queries to specify and filter data in BC4J entity objects
- ◆ View attributes correspond to columns of the query result



View Links

- ◆ Represents relationship between query result sets of two view objects
- ◆ Used to express relationships between view objects
 - Only implemented in one direction
- ◆ Can be created independently of associations
 - No programmatic connection with associations



Application Modules

- ◆ Instantiate BC4J view objects and view links
- ◆ Contain definition of database connection used for data access
- ◆ Can have multiple modules within the same BC4J layer
- ◆ Provide view usages required by application
- ◆ Specify the way in which view usages are related by view links.
- ◆ Relationships are represented by a tree (application data model)





Physical Structure

BC4J Files

- ◆ XML – redundantly stores data structure of relational database
- ◆ Java classes – contain complex logic and application calls used by other parts of application
- ◆ Java library – provided by Oracle to handle JDBC generation



How BC4J Works

- ◆ Wizards generate XML files that define data structure
 - All file manipulation should be handled using the wizards
- ◆ Wizards generate Java classes to control what happens to the data
- ◆ Portable interface to database
 - Can add validation and business rules





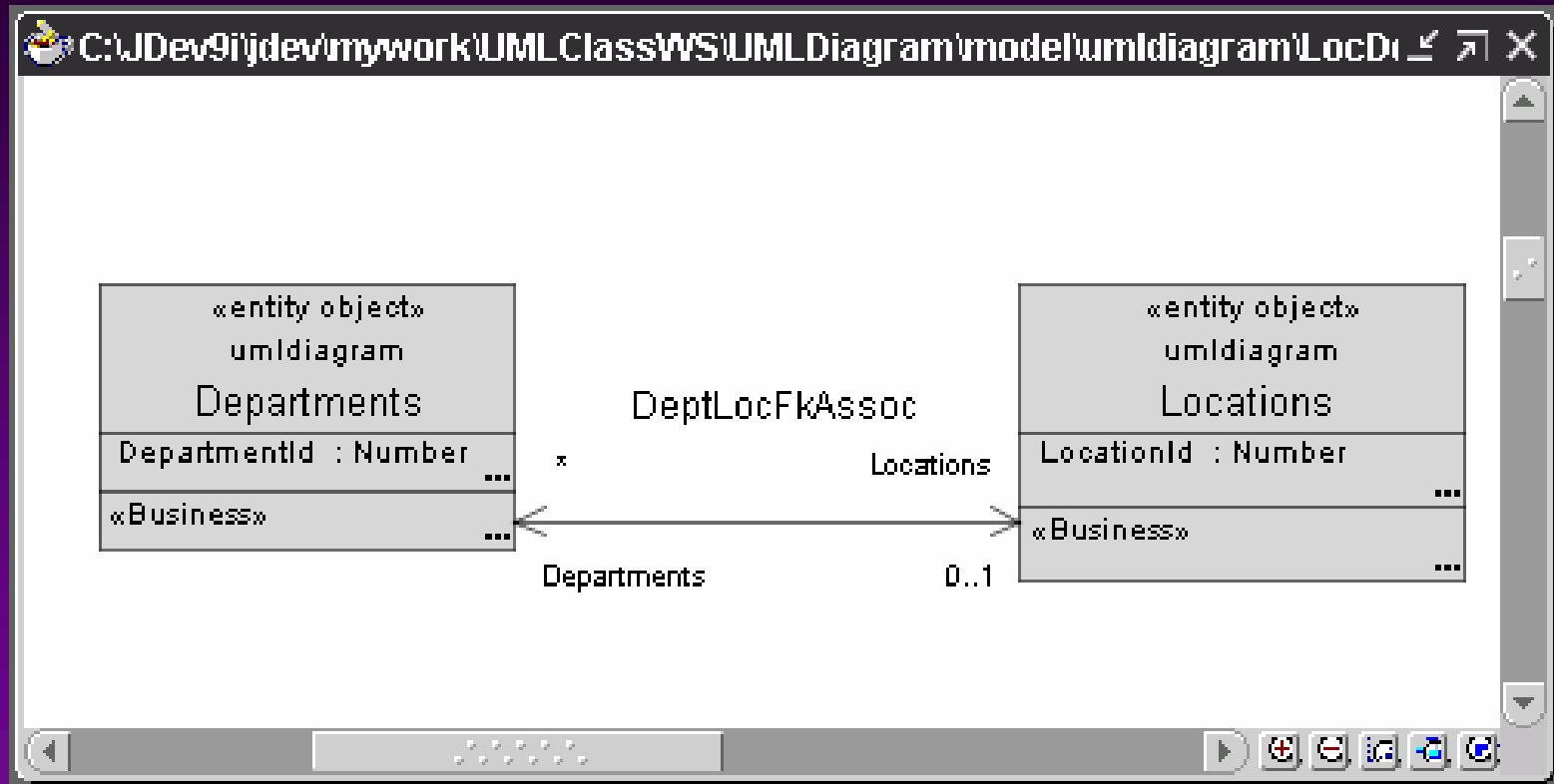
BC4J Capabilities

- ◆ Several options for placement of complex validation
 - Database tier (may impact performance)
 - Middle tier (application or web server)
 - Locally in the client (faster performance, easier debugging and maintenance if code not duplicated elsewhere)
- ◆ Factors to help determine placement
 - Hardware configuration
 - Deployment strategy
 - Number of and types of clients
 - Ease of debugging and maintenance



Using UML to Build BC4J Entity Objects

JDeveloper Class Diagram





UML Modeling

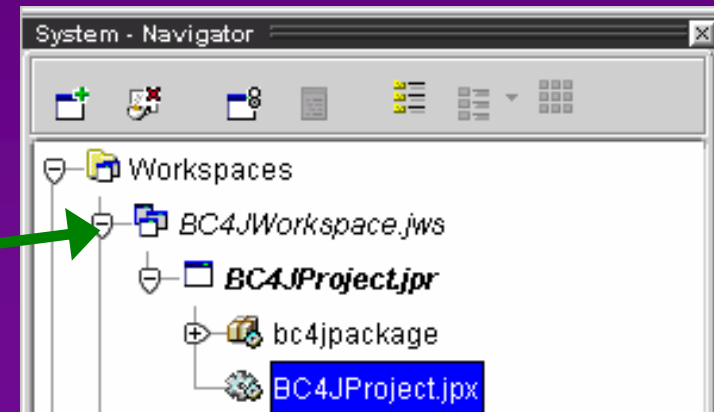
- ◆ Java classes
 - Quite good
 - Useful?
- ◆ BC4J
 - Only entity objects (not view objects)
- ◆ Data Modeling
 - Use Dulcian's BRIM™



BC4J Modeling

<u>UML</u>		<u>BC4J</u>
◆ Classes	→	Entities
◆ Attributes	→	Attributes
◆ Associations	→	Associations
◆ Generalization	→	Nothing??
◆ Interface	→	Nothing

Tip: Look at the classes that are italicized in the Navigator to see what was modified.



Features of the BC4J Class Modeler

Cool things

- ◆ 1. Entity class can be created from entities (drag and drop).
- ◆ 2. Entity class and entity can be created from tables (drag and drop).
- ◆ 3. On-screen editing
- ◆ 4. Table generation

Not cool things

- ◆ 1. No validation on in-place editing
- ◆ 2. Regeneration of tables drops the table.
- ◆ 3. It's only entities and associations.

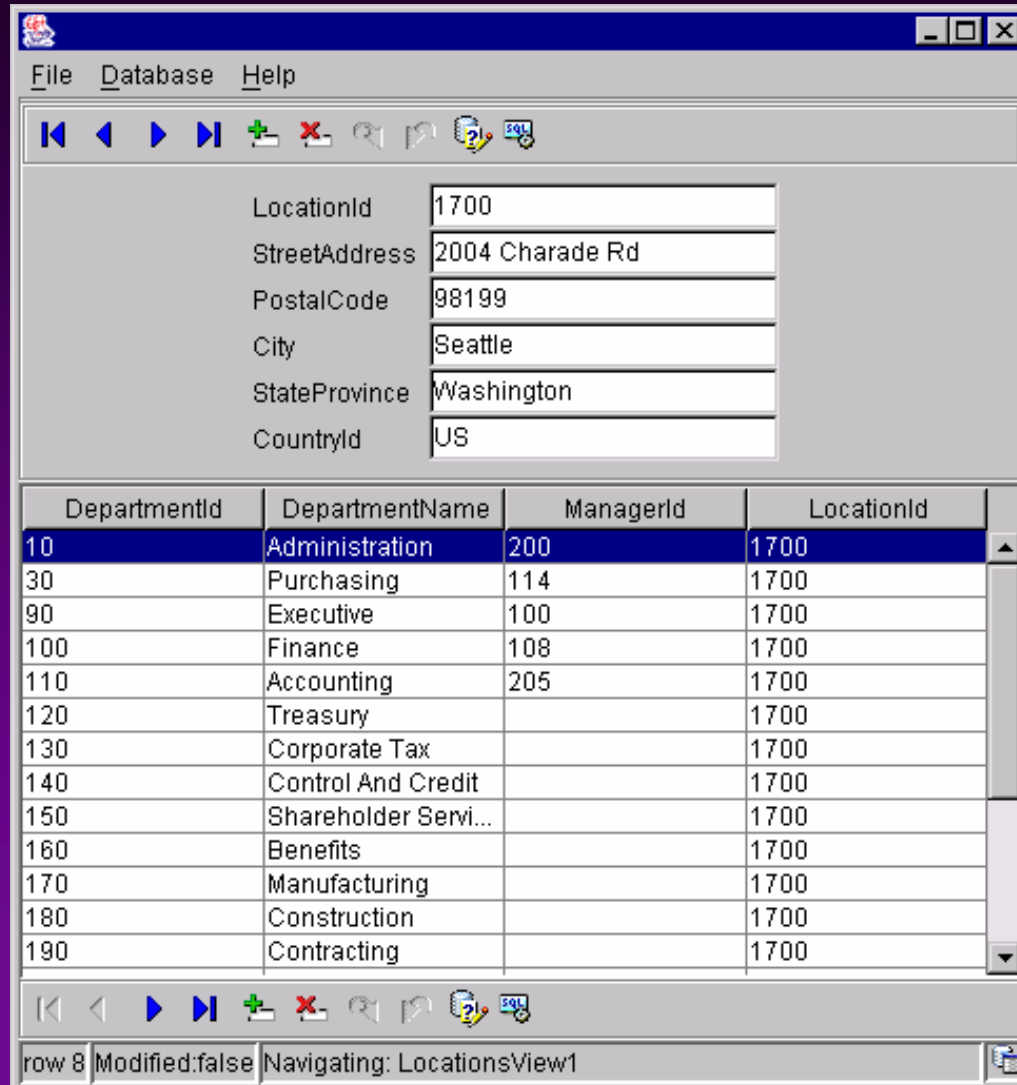


Building Applications with BC4J

New BC4J Features

- ◆ Synchronization with the database
- ◆ Bi-directional view links and many-to-many view links
- ◆ Create a view object based on a programmatic source
- ◆ Support for VARRAY and nested tables
- ◆ Design time support for mapping an attribute to a database sequence





The screenshot shows a Java Swing window titled "Nicer BC4J Tester". The window has a menu bar with "File", "Database", and "Help". Below the menu bar is a toolbar with icons for navigation and data operations. The main area is divided into two sections: a form for data entry and a table view.

The form contains the following fields:

- LocationId: 1700
- StreetAddress: 2004 Charade Rd
- PostalCode: 98199
- City: Seattle
- StateProvince: Washington
- CountryId: US

Below the form is a table with the following data:

DepartmentId	DepartmentName	ManagerId	LocationId
10	Administration	200	1700
30	Purchasing	114	1700
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700
120	Treasury		1700
130	Corporate Tax		1700
140	Control And Credit		1700
150	Shareholder Servi...		1700
160	Benefits		1700
170	Manufacturing		1700
180	Construction		1700
190	Contracting		1700

At the bottom of the window, there is a status bar showing "row 8 Modified:false Navigating: LocationsView1".

Building Applications Using BC4J

- ◆ Applications can be built quickly by connecting user interface to BC4J objects
- ◆ BC4J components provide alternative locations for logic
- ◆ Similar to having a virtual database
- ◆ BC4J provides an extensible framework as a basis for development



Using BC4J

- ◆ Application-specific BC4J
 - Builds set of Java classes to wrap DML commands
 - References an XML document that stores the data structure definition
- ◆ Extends base BC4J library classes using Java classes
- ◆ Encapsulates database access logic
- ◆ Implements complex data validation
- ◆ Caches shared data providing performance benefits



Using BC4J Components

- ◆ Conceptually divide application into 2 parts
 - User interface/client logic
 - Database interface/business logic
- ◆ Benefits
 - Helps simplify and organize projects
 - Optimize performance and code reusability
 - Improve maintainability by moving shared code into BC4J layer
 - Portability - Can be deployed on database server, application server or locally on the client

BC4J Architecture

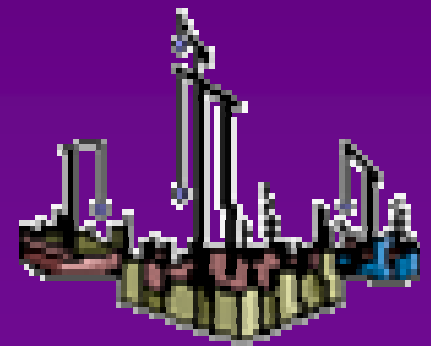
- ◆ Four architectures for business component clients:
 - JClient – architecture for creating Java GUIs
 - Thin Java Server Page (JSP) clients
 - XSQL clients – generate XML
 - UIX – framework for developing web applications





Building Java JClient Applications

- ◆ Bolt app module to UI project
 - JClient data model
- ◆ Bind to SWING components
- ◆ Any SWING/AWT can be used





Development Strategies

Approaches for using BC4J (1)

- ◆ 1. Database-centric approach
 - Build a small BC4J project for each program
 - Includes entity objects and associations with 1-1 relationship with tables, views and FK references
 - Create default view objects and view links for each entity objects and association
 - Assemble into application module
 - Modules kept relatively small
 - Add validation logic to entity objects
 - Each program has its own BC4J project



Database-Centric Approach: Pros and Cons

◆ Advantages

- Most comfortable for existing Oracle developers
- System not closely tied to JDeveloper and BC4J
- Most BC4J work can be supported using the JDeveloper wizards

◆ Disadvantages

- Ignores power and flexibility of BC4J
- Does not support BC4J reuse
- Does not take advantage of BC4J cache
- Does not take advantage of BC4J support of Web Tier and Business Tier servers
- Application heavily tied to Oracle DBMS
- Need skilled PL/SQL developers

Approaches for using BC4J (2)

◆ 2. Business-Logic Tier Approach

- Logic contained in Java class files residing outside of the database
- Create only one BC4J package
- Build BC4J application module for each program
- For large model, need to partition entity objects and associations into smaller subsets
- Entity objects can be shared by many different packages



Business Logic Tier Approach: Pros and Cons

◆ Advantages

- BC4J caching, project reuse, database independence
- Affords many development efficiencies
- Uses BC4J's ability to offload activity from database server

◆ Disadvantages

- Conceptually difficult approach
 - Business rules may reside in different places
- Rules should be carefully formalized
- Additional flexibility may create a system where bugs are difficult to locate due to logic residing in multiple places



Share your Knowledge: Call for Articles for the SELECT Journal

- ◆ Help contribute your knowledge to the larger Oracle community:
 - Make the SELECT Journal an even more valuable resource.
 - Articles wanted on topics of interest to the Oracle community.
 - Sign up to be a reviewer of articles.
- ◆ Submit articles, questions, ... to select@ioug.org.





Contact Information

- ◆ Dr. Paul Dorsey – paul_dorsey@dulcian.com
- ◆ Dulcian website - www.dulcian.com



Coming by year end
Oracle 9i JDeveloper Handbook

