



IBM Software Group

# IBM® Rational® Application Developer V6

## *EJB™ Mediator In Application Developer*



@business on demand.

© 2004 IBM Corporation  
Updated December 9, 2004

This presentation will focus on the Enterprise JavaBean (EJB) mediator in Application Developer.

## Goals

- Introduce tool support for the EJB Mediator technology
- Describe and demonstrate how this technology integrates with JavaServer Faces



The goals for this presentation are to introduce tool support for the EJB Mediator technology and to describe and demonstrate how this technology integrates with JavaServer faces.

## Agenda

- EJB Mediator

The agenda for this presentation is to look at the EJB Mediator in Application Developer.

## EJB Mediator Tool Overview

- EJB Mediator technology is available via the Create Session Bean Façade wizard
- Implemented using annotation based programming
  - ▶ Wizards generate appropriate annotations
  - ▶ User may also provide annotations
- Add session façade to a JSF page
  - ▶ Select EJB Session Bean from Data drawer or Page Data view



In Application Developer the EJB mediator technology is available via the Create Session Bean Facade wizard. This technology is implemented in application developer using annotation based programming. When using the Create Session Bean Facade wizard, application developer will automatically generate the appropriate annotations in the CMP Entity bean file. You can also provide your own custom annotations to this file.

The EJB mediator technology has also been integrated with JSF page development. JSF developers can add a session bean facade to a JSF page from the data drawer for the page data view by selecting an EJB Session Bean as their data source.

IBM Software Group IBM

## Creating a Session Bean Façade

Create Session Bean Façade menu option

Specifying the name and package for the new façade

Choose appropriate CMP beans

SDO Name

Read Only

Choose CMP and CMR fields to be included in SDO

EJB Mediator In Application Developer © 2004 IBM Corporation

This slide illustrates the Create Session Bean Façade wizard. To invoke the Create Session Beans Façade wizard, right click on the appropriate entity EJB under the Deployment Descriptor in the Project Explorer. From the context menu, select Create Session Bean Façade. On the first page of the wizard you will be asked to specify the name and a package for the new session façade . You will also be asked to choose the appropriate CMP been for this session façade. On the second page of the wizard you will provide an SDO name as well as select the CMP and CMR fields to be included in the SDO DataGraph. You are also given the option to have this data object be read only and also whether or not any CMR entities are contained within the DataObject or within the root data object. The final page of the wizard (not shown here) provides you with the opportunity to include this data object in a new or existing class diagram.

## Session Bean Façade: Code Updates

Session Bean added to the deployment descriptor

```

package com.ibm.websphere.samples.bank.ejb;

import java.util.ArrayList;

/**
 * @ws.sbf.session-facade
 *   name="CustomerFacade"
 *   value-objects="Customer"
 * @ws.sdo.value-object
 *   name="Customer"
 *   read-only="false"
 * Bean implementation class for Enterprise Bean: Customer
 */
public abstract class CustomerBean implements javax.ejb.EntityBean {
    private javax.ejb.EntityContext myEntityCtx;
  
```

Annotations are added to the Customer CMP Entity Bean

This slide illustrates the code and code artifacts that are on added as a result of running the Create Session Beans Façade wizard. Notice under the deployment descriptor in the project explorer view there is a new CustomerFacade session bean available. Also, if you open the appropriate CMP entity bean (in this case the Customer bean) you will see the annotations that have been added to support the CustomerFacade for this entity bean.

IBM Software Group IBM

## Session Bean Façade: Auto-Generated Code

The screenshot displays the IBM Application Developer interface. The Project Explorer on the left shows a project structure with folders for 'Session Beans', 'gen/src', and 'sdo'. The Outline view in the center shows the 'CustomerFacadeBean' class with methods like 'getSessionContext()', 'setSessionContext()', 'ejbCreate()', 'ejbRemove()', 'ejbActivate()', 'ejbPassivate()', 'getAllCustomerObjects()', 'getCustomerByKey()', 'createCustomer()', 'deleteCustomer()', 'applyCustomerRootChanges()', and 'getCustomerRoot()'. The Source view on the right shows the 'Customer' class with methods like 'getCustomerNumber()', 'setCustomerNumber()', 'getLastName()', 'setLastName()', 'getFirstName()', 'setFirstName()', 'getTaxID()', 'setTaxID()', 'getAccounts()', 'setAccounts()', and 'createAccounts()'. A red dashed box highlights the 'Customer' class with the label 'Static DataObject Interfaces'. A yellow callout box points to the 'gen/src' folder with the text 'SDO implementations classes'. Another yellow callout box points to the 'sdo' folder with the text 'SDO utility classes used to facilitate working with static SDO classes'. The bottom of the slide features the text 'EJB Mediator In Application Developer' and '© 2004 IBM Corporation'.

This slide illustrates the auto-generated code that is used to support the static SDO classes used by the CustomerFacade session bean. These classes can be found under the gen/src folder in the Project Explorer view. Of particular interest, is the static DataObject interfaces. Also notice the SDO implementation classes as well as the SDO utility classes used to facilitate working with the SDO classes.

Also notice on this slide the Outline view for the CustomerFacadeBean class. Notice the methods that have been added to this class. These methods enable users to query, create, update, and delete customers. Note that there are only two customer query methods automatically generated for the CustomerFacade; one that queries a specific customer based on the key or one that retrieves all customers. Support for adding custom queries is discussed on the next slide.

## Session Bean Façade: Custom Queries

- Supported via annotation based programming only
  - Annotations are added to the appropriate entity bean class

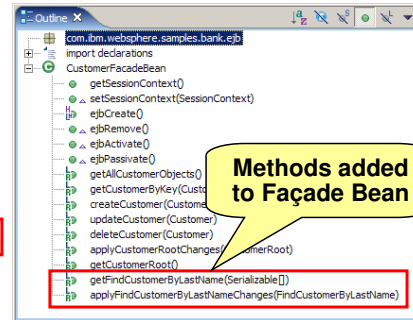
### Example:

```

/**
 * @ws.sbf.session-facade
 *   name="CustomerFacade"
 *   value-objects="Customer"
 *   queries="FindCustomerByLastName"
 * @ws.sdo.value-object
 *   name="Customer"
 *   read-only="false"
 * @ws.sbf.query
 *   name="FindCustomerByLastName"
 *   query="[select {$Customer} as c where c.lastName = ?1]"
 */

```

Annotations



Methods added to Façade Bean

This slide provides an example of how to provide a custom query on a session bean facade. This support is available using annotation based programming only and there is no wizard support for adding this type of query. To add a custom query, annotations must be added to the appropriate entity bean class. This slide provides an example of a custom query that has been added to a CMP entity bean class. Of particular interest are annotations that have been circled in red. Also included on the slide is the Outline view of the CustomerFacadeBean after the custom query had been added as shown in the example annotation to the left.



## Summary

- Application Developer supports the EJB Mediator through the Create Session Bean Façade functionality
- Tool integrates EJB mediator with JSF technology to enable Rapid Application Development



In summary, this presentation has focused on how Application Developer supports the EJB Mediator through the Create Session Bean Façade functionality and how this technology can be integrated with JSF support to provide Rapid Application Development capabilities when building J2EE applications.

## Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MOSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIK	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.