



IBM Software Group

IBM® Rational® Application Developer V6

JDBC™ Mediator in Application Developer



@business on demand.

© 2004 IBM Corporation
Updated January 25, 2005

This presentation will focus on the JDBC Mediator in Application Developer.

Goals

- Introduce tool support for the
 - ▶ JDBC Mediator
 - Relational Record
 - Relational Record List
- Describe and demonstrate how this technology integrates with JavaServer Faces (JSF) and JavaServer Pages (JSP) development



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

Agenda

- JDBC Mediator

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

JDBC Mediator Tools Overview

- Provides a Service Data Objects (SDO) Data Mediator to access data stored in a relational database
- Tool support simplifies Web application development
 - ▶ Integrated with JSF and JSP page development
 - ▶ Automatically generates appropriate data mediator and SDO code to access data
- Two types of Page Data are available
 - ▶ Relational Record
 - ▶ Relational Record List



The JDBC mediator provides an SDO data mediator to access data that is stored in relational database. In Application Developer, this support is available from the page data view as either a relational record order relation of record list. The relation of record is used to represent a row in relational database, were as a relational record list can be thought of as multiple rows resulting from a particular SQL query.

The JDBC mediator support in Application Developer simplifies web application development because of the tight integration with JSF and JSP page development. Application developer automatically generates the appropriate data mediator and SDO code to access relational data so that the developer does not need to code to these API specifically. It is this type of support that brings rapid application development capabilities to J2EE web-based programming.

JDBC Mediator Tools Overview (cont.)

- Migration support is available for V5.1.2 applications using WDO support
 - ▶ DataObjectAccessBean
 - ▶ DataListAccessBean
- } Replaced with SDO APIs

As you may recall WDO (precursor to SDO technology, that provided JDBC data access only) was first introduced into Application Developer in version 5.1.1 (Beta support) and 5.1.2 (GA support). At the time of these releases the SDO APIs were not yet available so Application Developer used an API that is different from the SDO APIs used in this release. Because of this, there have been several page code changes that JSF developers should be aware of. Specifically, the `DataObjectAccessBean` in the `DataListAccessBean` have been replaced with the specific SDO-based APIs. It is also important to note that there is support available for migrating WDO applications from V5.1.2 to the new APIs used in version 6. The details of the migration support can be found in the migration guide included with the Application Developer installation media.

Defining a New Relational Record / Record List

- Available from the Page Data view or from the Data drawer on the Palette
- A configuration wizard guides you through
 - ▶ Specifying the name for a new record or record list
 - Can re-use existing record or record list
 - ▶ Specifying initialization action (Relational Record only)
 - Fill in record with existing item in database or create empty record
 - ▶ Selecting / Creating a database connection
 - ▶ Selecting columns and building appropriate query



The Relational Record or Relational Record List is available from the Page Data view or from the Data drawer on the palette. These data items are available for JSF or JSP development. When adding Record or Record List to a JSF/JSP page, a configuration wizard guides you through several steps. In the first step you specify the name of a new or existing Record or Record List. If you are creating a new Relational Record you can specify initialization action that indicates whether or not to fill the record with an existing item in the database or to create an empty record. The configuration wizard also guides you through selecting an existing or creating a new database connection. Once this is complete you can proceed by selecting the appropriate columns and building the required query for your Relational Record or Record List.

Defining a New Relational Record / Record List

- As part of the creation process Application Developer does the following
 - ▶ Generates access code in the appropriate page code file
 - ▶ Makes query results accessible for binding with JSF UI components from the page data view
 - ▶ Generates or updates the following artifacts
 - Metadata XML file
 - Connection configuration file
 - Web deployment descriptor
 - Enhanced EAR configuration

When you create a new Relational Record or Record List there are a number of things that the tool does as part of this creation process. First, if the record or record list is associated with a JSF page, the tool will generate access code in the appropriate page code file. After creating a relational record or record list the results are available in the page data view and can be added to your JSF/JSP page if this was not already done as part of the creation process. You should also be aware that the tools automatically generate or update the following artifacts in your project : the metadata XML file, connection configuration file, Web deployment descriptor, and the enhanced EAR file.

Demo: Create a New Relational Record

- This demonstration will highlight
 - ▶ Creating a new Relational Record (for update)
 - ▶ Resources created by Application Developer to support a relational record and record list
 - ▶ Code added to the page code file



From the slide you can access a demonstration that will highlight how to create a relational record for update in association with a JSF page. This demonstration will also point out some of the key resources created by application developer and the code that is added to the page code file for the associated JSF page.

Adding Controls: Relational Record / Record List

- To add JSF controls for a record or record list
 - ▶ Drag record/record list from Page Data onto JSF page
 - ▶ Right click on record/ record list in Page Data view and select “Insert new controls for...”

- Controls for a record
 - ▶ Input / Output

- Controls for a record list
 - ▶ Multi valued table
 - ▶ Single valued table

Once you've added a record or record list to the page data for a JSF or JSP page the next step is to add UI controls to the page associated with this data. This slide primarily discusses this with respect to JSF development. From the page data view there are two ways to add controls to the page associated with this data. The first is to drag record/record list from Page Data onto JSF page. The second option is to right click on record/ record list in Page Data view and select “Insert new controls for...”. Your options for the types of data controls will depend on whether the data is a record or record list. For records you can choose either input or output controls depending on whether you are planning to update the data or simply display the data. For record lists, you can choose to display the data in a multi valued table or as a single valued table.

Handling Updates: Relational Record / Record List

- Use Quick Edit view for command components to add action code automatically
- Write custom code using SDO APIs



If the record or record list is associated with a JSF page you can add code to the appropriate action method (associated with a command component) to handle updates to the relational data. You can do this from the Quick edit view and use the appropriate code snippets from the context menus, or write custom code using the SDO and mediator APIs to perform the updates.

Demo: Updating a Relational Record

- This demonstration will highlight
 - ▶ Adding update controls for a relational record
 - ▶ Adding code to handle updates to relational record



From the slide you can access a demonstration that will highlight how to add update controls to a JSF page for a relational record as well as how to add code to handle updates as part of an action method.

Summary

- Application Developer supports the EJB Mediator through the Relational Record and Record List
- Tools integrate JDBC and EJB mediator with JSF technology to enable Rapid Application Development

In summary, this presentation has focused on how Application Developer supports the JDBC Mediator through the Relational Record and Record List page data items and how this technology can be integrated with JSF support to provide Rapid Application Development capabilities when building J2EE applications.

Trademarks and Disclaimers

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

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

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

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.

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.

