



IBM Software Group

IBM WebSphere Application Server V6

Dynamic query



@business on demand.

© 2004, 2006 IBM Corporation
Converted to video May 13, 2015

This presentation will discuss the Dynamic Query feature of IBM WebSphere Application Server V6. Dynamic Query allows you to dynamically build and submit queries at runtime.

Goals

- Understand dynamic query feature and the benefits
- Understand how to use dynamic query in your applications



The goals of this presentation are to gain an understanding of Dynamic Query and how to take advantage of the feature in your applications.

Agenda

- Overview
- Using dynamic query
- Big picture
- Summary



The agenda for this presentation includes an overview of Dynamic Query and a brief description of how to use Dynamic Query.

Section

Overview



The next section will provide an overview of Dynamic Query, the advantages of using Dynamic Query, and a brief discussion on how to use Dynamic Query.

Dynamic query service: Overview

- Allows formulating and running Enterprise Java™ Bean Query Language (EJB QL) queries
 - ▶ As opposed to standard EJB QL specified at application development time in the deployment descriptor
 - ▶ Reduces need to create many different finders
- Many extensions to the EJB QL syntax:
 - ▶ Invoke value object or bean methods
 - ▶ Return multiple CMP or CMR fields in a single SELECT
 - ▶ Return a computed expression in the query



Dynamic Query allows you to dynamically build and submit queries that select, sort, join, and perform calculations on application data at runtime. Dynamic Query Service provides the ability to pass in and process EJB QL queries at runtime, eliminating the need to specify required queries into the deployment descriptors during development.

EJB Query Language is defined by the Enterprise JavaBean 2.1 specification. EJB QL provides a standard way to implement portable finder methods and simple select methods that can be used to either select sets of EJB references or attribute data. WebSphere Application Server's Dynamic Query provides extensions to EJB QL, such as the ability to invoke EJB bean methods or value object methods, use multiple element select clauses, and the use of **group by** and **subselect** in your queries.

Dynamic query service: Overview (cont.)

- Enables a variety of different capabilities
 - ▶ Interactive testing of EJB queries without redeployment to update a finder or select query
 - ▶ Fine-grained selection and grouping criteria, including method invocations, eliminates the need to write complex Java code and complex grouping and sorting logic for query results
- Standard J2EE applications can be augmented with Dynamic Query functionality
 - ▶ No change to the existing EJB model, implementations or their deployments
 - ▶ Existing applications remain compliant
 - ▶ New query functionality only available on the WebSphere Application Server platform



Dynamic Query enables a number of different capabilities. For example, you can run interactive testing of your queries without the need to redeploy your EJB application. Fine-grained selection and grouping criteria help to eliminate the need for complex Java code and complex grouping and sorting logic for your query results. This can reduce development time. Existing J2EE applications can take advantage of dynamic query very easily in a non-intrusive manner because there is no change required to the existing EJB model, implementations, or their deployments. Application clients will use the query service to dynamically invoke specified queries against application business objects.

Using the dynamic query service

- Service exposed as a stateless session EJB in query.ear
 - ▶ “Query bean” needs to be looked up and used according to the standard EJB client programming
 - ▶ The dynamic “query bean” has both a remote and a local interface
- Two options for client methods
 - ▶ *executeQuery()*
 - ▶ *prepareQuery()* and *executePlan()*
- Security follows standard J2EE authorization scheme



Dynamic Query service is implemented as a separate J2EE application that is installed on the server and works directly with the EJB Container to obtain schema information when executing queries. The dynamic “Query Bean” is packaged as the query.ear file in the installableApps directory of the WebSphere Application Server. The system administrator may need to install this ear into the Application Servers. In the single server environment, installation is automatic.

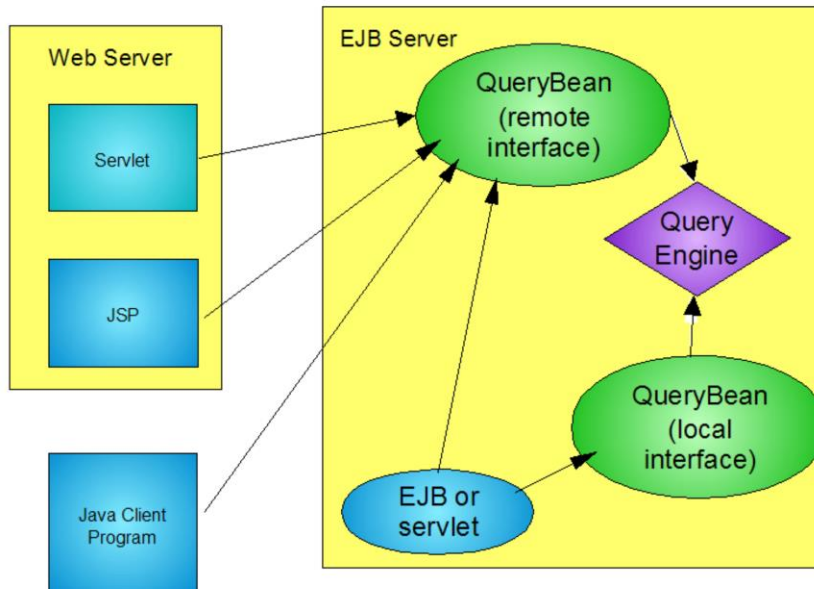
The remote interface allows for retrieval of large query results in pieces by specifying skipRow and startRow limit parameters.

The local interface supports "demand driven" iterators. Data is returned incrementally. For each record retrieval from the database, the next() method must be called on the iterator.

Application clients will be interacting with “Query Bean” which is a stateless session bean. There are two options – invoking executeQuery(), or invoking prepareQuery() and executePlan(). Basically, executeQuery is a combination of prepareQuery and executePlan. In repetitive types of situations, you may want to use prepareQuery() and executePlan() to redistribute the processing for better application performance. The prepareQuery() method parses and translates your query and returns a string called a query plan. Save this string in your application, and then call the executePlan() method as many times as required, passing in the previously saved string to run your query. In this case, you avoid preparing the query again and again unnecessarily.

As stated, dynamic query service works directly with the EJB Container and not your application when obtaining schema information and executing queries. Therefore, any J2EE authorization you may have configured within your application, will not take precedence on the dynamic query service. If security is a concern, you also need to configure it on the dynamic query service application.

Dynamic query interfaces



Here you see an example of the “big picture” view of the Dynamic Query Interfaces. The query bean has both a remote and a local interface. The local interface may be used for access from clients within the same application server, and the remote interface must be used for access from clients running outside of the application server.

Section

Summary

Summary

- Understand dynamic query and it's benefits
- Understand how to enable dynamic query functionality on the application server
- Learned how to invoke dynamic query functionality from your application



In this presentation you learned about the benefits of the Dynamic Query function, how to enable the use of Dynamic Query on the Application Server, and how to invoke Dynamic Query from your application.

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	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e (logo)/business	DB2	Series	OS/400	xSeries
AX	DB2 UniversalDatabase	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 NONINFRINGEMENT. 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 2006. 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.

