



IBM Software Group

IBM® WebSphere® Application Server V7

DB2® and WebSphere integration



@business on demand.

© 2008 IBM Corporation
Updated September 23, 2008

This presentation discusses DB2 and WebSphere integration for WebSphere Application Server V7.0 on z/OS®.

Agenda

- SQLj deployment in WebSphere Application Server
- PureQuery support in WebSphere Application Server
- DB2 trusted context support
- Extended DB2 data source support
- DB2 client reroute enhancements in WebSphere Application Server

WebSphere Application Server version 7.0 includes three new features to take better advantage of DB2.

SQLj offers great performance advantages over JDBC, especially when used in the z/OS environment and in most cases SQLj is recommended for WebSphere applications on z/OS when running DB2 on z/OS as the backend Database. With previous versions of WebSphere, however, it was difficult and time consuming to go through the typical tasks of installing, deploying, customizing, and binding SQLj applications since earlier versions of WebSphere application Server administrative console (or scripting) does not offer a way to perform these tasks. WebSphere Application Server V7 supports these tasks

WebSphere Application Server V7 supports new features in DB2 V9.1. With DB2 trusted contexts, you can preserve end-user identity when interacting with the database for accountability reasons without much effort, strengthen database security by avoiding granting all privileges to a single user, and allow for client identity propagation without the performance penalty of having to close and reopen connections with a different identity.

WebSphere Application Server V7 will integrate DB2 client reroute by allowing you to easily configure DB2 client reroute. WebSphere also offers the option to persist the Client Reroute data such that if the application server locks up after a reroute happens, it will not try to connect to the old DB2 Server. Instead, the application server will persist the clientReroute information, which is updated by DB2 at runtime, and allow DB2 to use it on the next connection - assuming the original DB2 connection fails.

Section

SQLj deployment in WebSphere



This section covers SQLj deployment in WebSphere.

SQLj deployment in WebSphere

- Prior steps required to configure SQLj
 - ▶ Deploy your application stand-alone
 - ▶ Un-jar the deployed ear file from first step.
 - ▶ Customize and bind the .ser files that exist in the ear file
 - ▶ Re-jar the ear file.
 - ▶ Finally, reinstall the ear file with no deploy option
- WebSphere V7 enhancement
 - ▶ Installation of SQLj applications from WebSphere Application Server administrative console.
 - ▶ Customization and binding of DB2 SQLj profiles from administrative console without the need to uninstall and reinstall applications

Before WebSphere Application Server V7, it was difficult and time consuming to go through the tasks of installing, deploying, customizing, and binding SQLj applications, since earlier versions of the administration console and scripting do not offer a way to perform these tasks. Instead, you were forced to perform the steps listed here. This a labor intensive process and if, for example, you forget and redeploy when you reinstall the .ear file, you would lose all the work you did in getting to this point and need to start over.

IBM Rational Application Developer provides a way to do the binding and customization of the .ear file, but that is not quite enough, since Rational Application Developer is not the WebSphere Application Server administrative console. When the application is installed on WebSphere Application Server, you can make the changes in the administration console or scripting environment without needing to uninstall and reinstall.

This process is greatly streamlined with WebSphere Application Server V7. First, customization of DB2 SQLj profiles is supported in the administration console and scripting environments. Thus, administrators are able to recustomize and rebind the SQLj profiles and SQLj application right from the administrative console and without the need to uninstall and reinstall the application.

By default, the processing of an SQLj profile that contains embedded SQL statements creates four DB2 packages in the database; one for each isolation level. The customization process augments the profiles with DB2-specific information for use at runtime. Typically, customization is done after the SQLj application has been translated and before the application is run. Omitting the customization step will cause the SQLj application to use dynamic SQL (that is, just like JDBC).

Integrated Solutions Console : SQLj

The screenshot displays the 'Enterprise Applications' console for the application 'PlantsByWebSphere'. The 'General Properties' section shows the application name and validation options. The 'Detail Properties' section includes 'Target specific application status'. The 'Database Profiles' section on the right contains a link for 'SQLj profiles and pureQuery bind files', highlighted in yellow with a red arrow pointing to it.

An inset window titled 'Install New Application' is shown, detailing the 'Provide options to perform the EJB Deploy' step. It lists various deployment options and their 'Enable' status. The 'Deploy EJB option - Database access type' field is highlighted with a red box and contains the value 'SQLj', which is also circled in red. The 'SQLj Classpath' field is empty.

At the bottom of the console, the text 'DB2 and WebSphere integration' and '© 2008 IBM Corporation' is visible, along with a page number '5'.

When installing an application with SQLj, set the “Database access type” field to SQLj, the options for this field are JDBC and SQLj as shown in this picture. The default for the access type is JDBC. Also the SQLj application now shows up on the main application page and will look for any SQLj profiles (.ser files) if clicked.

Selecting this item produces the entry panel on the next slide.

Integrated Solutions Console : SQLj

Integrated Solutions Console - Microsoft Internet Explorer

Address http://localhost:9079/bm/console/login.do

Integrated Solutions Console Welcome Help | Logout

SQLj profiles

Specify customization and binding settings for the DB2 SQLj profiles that are included in this application. SQLj profiles for other database types may be viewed but not changed.

Tasks (DB2 only)

- Customize and bind profiles
- Bind packages

SQLj profiles

- Select and order profiles to customize/bind

Available profiles

- AccessDeptEjb.jar\...\AccessDeptEjb\...\FS_DEPT_SJProfile3.ser
- AccessDeptEjb.jar\...\AccessDeptEjb\...\FS_DEPT_SJProfile4.ser
- AccessDeptEjb.jar\...\AccessDeptEjb\...\FS_DEPT_SJProfile5.ser
- AccessEmpEjb.jar\...\AccessEmpEjb\...\FS_SAMPLE_SJProfile0.ser
- AccessEmpEjb.jar\...\AccessEmpEjb\...\FS_SAMPLE_SJProfile1.ser

Add Remove

Selected profiles

- AccessDeptEjb.jar\...\AccessDeptEjb\...\FS_DEPT_SJProfile0.ser
- AccessDeptEjb.jar\...\AccessDeptEjb\...\FS_DEPT_SJProfile2.ser

Move Up Move Down

Customize/bind the selected SQLj profiles as a group

Use a profile group file to specify profiles to customize/bind

The full path to profile_group file (.grp) Browse

Field help

This option creates a .grp file that contains the SQLj profiles that are selected. An option is provided on the next panel to download the .grp file.

Page help

[More information about this page](#)

Command Assistance

[View administrative action command for last action](#)

6

DB2 and WebSphere integration © 2008 IBM Corporation

This panel is used for binding and customizing an SQLj profile.

Section

PureQuery support



This section covers pureQuery support in WebSphere.

PureQuery support

PureQuery is a high-performance Java data access technology focused on simplifying the tasks of developing and managing applications that access data.

Improves the Java data access life cycle

- ▶ Existing and New applications
- ▶ Tools, APIs, and runtime environment
- ▶ Single API to query databases and in-memory Java objects
- ▶ Embraces SQL as the common query language
- ▶ Simplify SQL data access

The benefits of using pureQuery extend throughout the development, deployment, management, and governance stages of the application life cycle.

pureQuery provides access to data in databases and in-memory Java objects using its tools, APIs, and runtime environment.

The Java data access life cycle is improved during the development, deployment, management, and governance stages. pureQuery is a key component of IBM's end-to-end problem determination strategy.

Before pureQuery, JDBC or SQLj was used to access a database. This was tedious to develop, customize, and maintain. A developer could spend a lot of time on lower level data access instead of business logic. Technologies based on proprietary query languages provided no visibility to the efficiency of the generated SQL. Problem determination was difficult. The proprietary query language was tied to a single vendor. These technologies were not sophisticated enough to handle complex scenarios. The performance boost from Static SQL required new skills. There was no simple integrated way to work with database and in-memory Java objects.

With pureQuery, you can use standard SQL to access data from databases or in-memory Java objects.

Section

DB2 trusted context



This section covers DB2 trusted context in WebSphere.

DB2 trusted context

- A trusted context is an object the database administrator defines that contains a system authorization ID and a set of trust attributes.
- The relationship between a database connection and a trusted context is
 - ▶ established when the connection to the database server is first created
 - ▶ relationship remains for the life of the database connection.



A trusted context is an object that the database administrator defines that contains a system authorization ID and a set of trust attributes. Currently, for DB2 database servers, a database connection is the only type of context that is supported. The trust attributes identify a set of characteristics of a connection that are required for the connection to be considered a trusted connection. The relationship between a database connection and a trusted context is established when the connection to the database server is first created, and that relationship remains for the life of the database connection. After a trusted context is defined, and an initial trusted connection to the DB2 database server is made, the middleware server can use that database connection under a different user ID without re-authenticating the new user at the database server - assuming that the trusted context is created without the authentication option being required.

A DB2 trusted context with authentication provides an option where a password is required when switching identity. Plug-in points for trusted context support with authentication allow you to plug-in your own code to implement a trusted context with authentication.

DB2 trusted context

- Allows the use of the trusted DB2 Connection under a different user without reauthenticating the new user at the database server
 - ▶ The trusted context must have been created without the authentication option being required
- Client identity propagation removes the performance penalty of having to close and reopen connections with a different identity



Assuming the trusted context is created without the authentication option being required, WebSphere Application Server is allowed to use the trusted DB2 Connection under a different user ID without reauthenticating the new user ID at the database server. Client identity propagation eliminates the performance penalty of having to close and reopen the connections with a different identity.

Section

Extended DB2 data source support



This section covers SQLj deployment in WebSphere.

Extended DB2 data source support

- Adds a great level of flexibility to administrators who are running against DB2 using the DB2 universal jdbc driver.
- Heterogeneous pooling allows WebSphere administrators to configure a WebSphere DB2 data source
 - ▶ using a set of core datasource properties
 - ▶ defer to each application to define application specific non-core datasource properties.



Extended DB2 data source support, also known as heterogeneous pooling or HP, allows you to easily configure a WebSphere DB2 data source with a set of core data source properties. Application-specific, non-core data source properties are deferred to each application to define.

An application can define one or more non-core sets of data source properties and associate each set with a different resource reference that points to the main WebSphere DB2 data source.

This feature allows applications to share the same WebSphere connection pool even though each application might have its own set of different data source properties.

The sharing will lead to a reduction of the number of open connections, which in turn reduces resource consumption on both the client side - WebSphere - and the server side - the database layer.

Extended DB2 data source support

- Heterogeneous pooling allows
 - ▶ applications to override non-core data source custom properties
 - ▶ sharing connections between different resource references that have the same core properties, within same transaction, avoiding two phase commit if one and only one connection is used.



In the application, non-core data source custom properties can be overridden using heterogeneous pooling.

In heterogeneous pooling, an option is provided to allow sharing connections between different resource references that have the same core properties within the same transaction, which leads to the avoidance of two-phase commit processing if the sharing leads to one and only one connection being used.

Extended DB2 data source support

[JDBC providers](#) > [DB2 Using IBM JCC Driver](#) > [Data sources](#) > [DB2 Using IBM JCC Driver - DataSource](#) > [WebSphere Application Server data source properties](#)

Use this page to set WebSphere(R) Application Server connection management-specific properties that affect a connection pool.

Configuration

General Properties

Advanced DB2 features

Optimize for get/use/close/connection pattern with heterogenous pooling

DB2 automatic client reroute options

Retry interval for client reroute
 seconds

Maximum retries for client reroute
 retries

Alternate server names

Alternate port numbers

Client reroute server list JNDI name

Unbind client reroute list from JNDI

Apply OK Reset Cancel



This panel is used to set WebSphere Application Server connection management specific properties that affect a connection pool.

Extended DB2 data source support

- Heterogeneous pooling (HP) benefits:
 - ▶ Reduction in memory consumption on WebSphere Application Server side
 - ▶ Reduction in number of open connections to database which leads to memory reduction on database side
 - ▶ Can increase performance



Heterogeneous pooling allows applications to share the same WebSphere connection pool even though each application might have its own set of different data source properties. The sharing results in a reduction in memory consumption on the WebSphere side. Many data source definitions can correspond to one connection pool, thus, there are fewer objects in memory in comparison with each data source definition corresponding to its own connection pool.

Secondly, the reduction in the number of open connections to the database results in using less memory on the database side. This can happen by reducing the number of idle connections. Having one pool correspond to multiple data source definitions causes more utilization of connections in comparison with a separate pool for each data source definition.

Performance is increased by allowing the avoidance of the use of two-phase commits in certain DB2 scenarios. As a result of the DB2 heterogeneous support, you can now hint to WebSphere to reuse connections even if they differ in properties, which will result in not using two-phase commit transactions. This will require the get→use→close pattern to take affect.

In some cases, the number of global transaction branches is reduced or even eliminated.

Section

DB2 client reroute



This section covers SQLj deployment in WebSphere.

DB2 automatic client reroute (ACR)

- Client reroute available on
 - ▶ Enterprise Server Edition with data partitioning feature
 - ▶ Data propagator-style replication
 - ▶ High availability cluster multiprocessor (HACMP™)
 - ▶ High availability disaster recovery (HADR)



Client reroute is a feature that enables a DB2 UDB client application to recover from a loss of communications so that the application can continue its work with minimal interruption. As the name implies, rerouting is central to the support of continuous operations. But rerouting is only possible when there is an alternate location that is identified to the client connection. Before WebSphere Application Server V7.0 you could not set the alternate server information in the administrative console. If alternate server information is used on first connection to the database when running with a type 4 JDBC driver, this same information is required for the reroute. DB2 JDBC driver Type 4 updates an in-memory reference to the alternate server information on the first successful connection so that if the DB2 server changes its alternate information, the client is kept up to date while the data source property contains the old information. WebSphere Application Server V7 adds a feature to permanently persist the in-memory object so that if the WebSphere Application Server restarts or stops, the application server is able to connect to the right DB2 server when it comes back up. That is, will not use the stale server information that is set on the data source.

DB2 automatic client reroute (ACR)

- Makes High Availability Disaster Recovery (HADR) failover almost transparent to the applications.
- Enables applications to recover from a communication loss ensuring the application can continue to work with minimal or no interruptions



A client rerouting enhancement persists alternative server information in the JNDI when using type 4. This allows a connection to the correct DB2 server even after a WebSphere Application Server hangs. The WebSphere Application Server “Test Connection” function includes client reroute support.

DB2 automatic client reroute (ACR)

- Upon primary failure, DB2 client automatically attempts to connect to the alternate DB2 server
 - ▶ upon success, the DB2 client will return SQLException with errorCode -4498 to the applications to indicate a successful redirect.
 - ▶ WebSphere Application Server maps -4498 to a Stale Connection Exception to allow the applications to retry their logic in a database independent manner



Upon primary failure, the DB2 client automatically attempts to connect to the alternate DB2 server. If successful, an SQL exception with error code -4498 results. This exception indicates a successful redirect has occurred. WebSphere Application Server interprets the -4498 error code and allows the applications to retry their logic in a database independent manner.

Summary

- SQLj deployment processing is now streamlined
- pureQuery uses standard SQL to access data from databases or in-memory Java objects
- DB2 trusted context improves security and performance
- heterogeneous pooling reduces memory and increases performance
- DB2 client reroute enables a DB2 UDB client to recover from a communication loss with minimal interruption



This presentation has reviewed DB2 and WebSphere integration for WebSphere Application Server V7.0 on z/OS.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASv7_DB2andWASIntegration.ppt

This module is also available in PDF format at: [../WASv7_DB2andWASIntegration.pdf](..WASv7_DB2andWASIntegration.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

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:

DB2 HACMP IBM WebSphere

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

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 or changes in the products or programs 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 (for example, 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 2008. 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.

