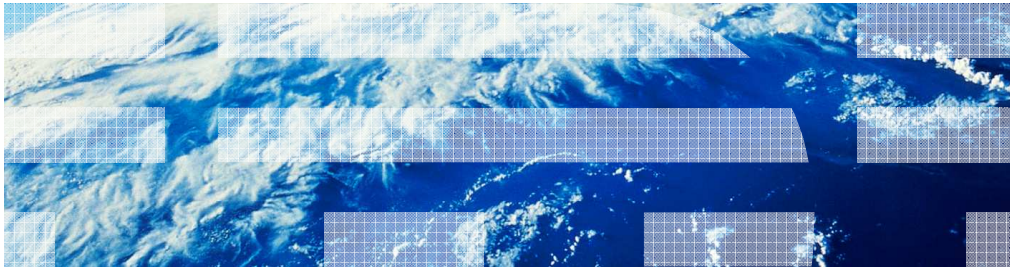

Troubleshooting database connection problems



This presentation will act as an introduction to database connection problems when using WebSphere Application Server V7.

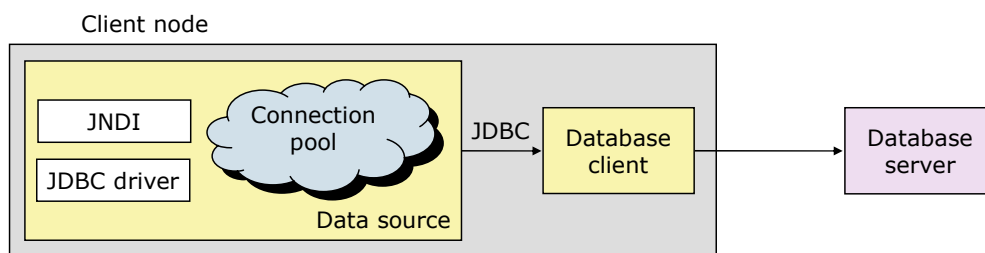
Unit objectives

- After completing this unit, you should be able to:
- Test data source connections to databases
- Describe J2EE Connector (J2C) authentication issues
- Describe Java Database Connectivity (JDBC) driver issues
- Use the built-in tester console to test connections
- Enable tracing

The goal of this presentation is to describe and understand issues that can occur with JDBC driver and J2C authentication, and be able to enable the proper tracing and use the built-in tester console to test database connections.

JDBC providers

- Provide the JDBC driver implementation for database access.
 - Type 2 JDBC drivers (thick)
 - Require the database client software on the client node to connect to the database server
 - Type 3 JDBC drivers (net protocol)
 - Require server side code to map net protocol to native database
 - Type 4 JDBC drivers (native protocol)
 - Connect directly to the database using its native protocol
- XA drivers support transaction recovery



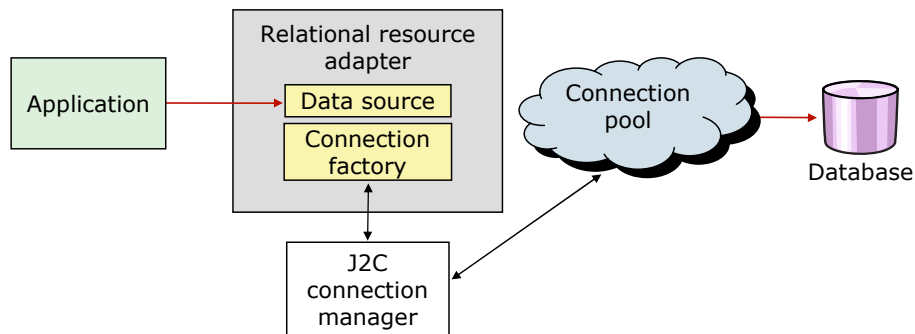
Introduction to database connection problems

© 2011 IBM Corporation

The JDBC provider object supplies the specific JDBC driver implementation class for access to a specific vendor database. To create a pool of connections to that database, you associate a data source with the JDBC provider. Together, the JDBC provider and the data source objects are functionally equivalent to the Java Platform, Enterprise Edition (Java EE) Connector Architecture (JCA) connection factory, which provides connectivity with a non-relational database.

Data sources

- Data sources can improve performance and portability for database access.
 - Standard and XA data sources
- Connection pooling is provided by two parts:
 - J2C connection manager
 - Relational resource adapter



Introduction to database connection problems

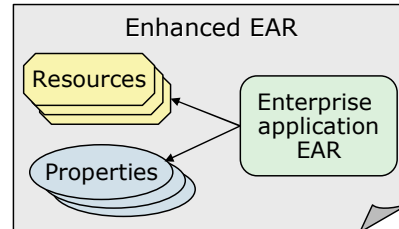
© 2011 IBM Corporation

A data source is associated with a JDBC provider, which supplies the driver implementation classes that are required for JDBC connectivity with your specific vendor database. Application components transact directly with the data source to obtain connection instances to your database. (Standard and XA data sources)

The connection pooling is provided by two parts, the J2C connection manager and Relational resource adapter, which corresponds to each data source provides connection management.

Enhanced EAR

- Enterprise archive containing Java EE artifacts plus resource information needed to install on WebSphere Application Server
 - JDBC resources (data sources)
 - Class loader
 - JAAS authentication aliases
 - Shared libraries
 - Virtual host information
- Support integrated with the IBM Rational Development and Assembly and Deployment tools
 - Found on the Deployment page of the application deployment descriptor
- **Warning:** Can possibly cause problems if unintended application scoped resources are used in production
 - Enhancements can be removed or ignored during application installation



An Enterprise Archive, or EAR, is a file format used by Java EE for packaging one or more modules into a single archive so that the deployment of the various modules onto an application server happens simultaneously and coherently. It also contains XML files called deployment descriptors which describe how to deploy the modules.

An enhanced EAR is an EAR file that also includes resource definitions for JDBC resources, class loaders, shared libraries, JAAS authentication aliases, and virtual host configuration. All of these configurations will be done at the Application scope rather than at the server, node, cluster, or cell level that the WebSphere administrator typically deals with. Application level resources are not viewable in the WebSphere administration console.

The role of the JDBC provider

- The JDBC provider object supplies the specific JDBC driver implementation class for access to a specific vendor database.
- The driver implementation classes are supplied by defining the WebSphere variables associated with the provider.
- Java classes and native libraries can be specified by the following:
 - DB2UNIVERSAL_JDBC_DRIVER_PATH
 - INFORMIX_JDBC_DRIVER_PATH
 - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH
 - ORACLE_JDBC_DRIVER_PATH
- Custom variables can be defined in WebSphere Application Server in order to accommodate any vendor-specific driver implementation

The JDBC provider object supplies the specific JDBC driver implementation class for access to a specific vendor database, such as DB2 V8. The driver implementation classes are supplied by defining the WebSphere variables associated with the specific provider. The variables reference both Java and native libraries. JDBC type one and two drivers require that JAR files and native libraries be referenced, while JDBC type three and four drivers only require JAR files to be referenced. Other custom variables can be defined in WAS in order to accommodate any vendor-specific implementation.

JDBC provider configuration

Configuration

General Properties

1 * Scope
cells:was7host01Cell01:nodes:was7host01Node01:servers:server1

2 * Name
DB2 Universal JDBC Driver Provider (XA)

Description
Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources

3 Class path
\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
\${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
\${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_licens
e_cisuz.jar

4 Native library path
\${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}

5 Isolate this resource provider

* Implementation class name
com.ibm.db2.jcc.DB2XADataSource

6

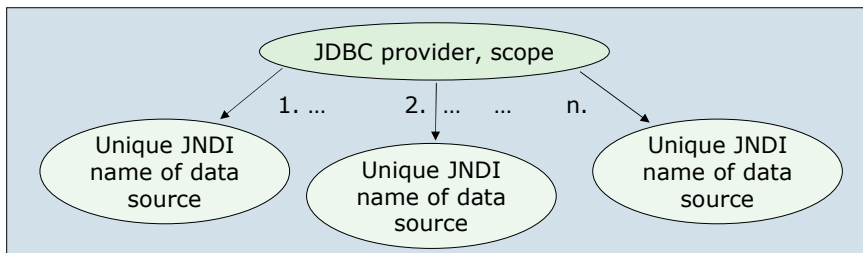
1. Scope
2. Name
3. Class path (Java)
4. Native path (C code)
5. Isolate provider (new V7)
6. Implementation class name

Introduction to database connection problems © 2011 IBM Corporation

This slide shows an example of a configured DB2 JDBC provider by way of the administrative console.

The role of the data source

- The JNDI name represents the runtime binding of the data source to the JDBC provider in a given scope.
 - A data source is analogous to a Java EE Connector Architecture (JCA) connection factory.



- Runtime binding creates a highly flexible environment but if problems exist, the symptoms are not seen until run time.
 - The test connection facility of the administrative console is helpful in uncovering such errors during the deployment phase

The data source in WebSphere Application Server V7 represents the actual connection to the database. A data source is comparable to a Java EE Connector Architecture (JCA) connection factory.

The data source uses the JDBC provider implementation to connect to the database system and relies on connection parameters such as the database name, the type of the driver being used, and the authentication credentials used to determine authorities and privileges for the connection. (The test connection facility of the administrative console is helpful in uncovering any errors that can happen during the deployment phase).

The authentication credentials are specified using a Java 2 Connection Architecture (J2C) data entry. This data entry is then specified during the creation of the data source and is used for all subsequent authentication.

Data source screen (1 of 3)

JDBC providers

[JDBC providers](#) > [DB2 Universal JDBC Driver Provider \(XA\)](#) > [Data sources](#) > **DataSource**

Use this page to edit the settings of a datasource that is associated with your application. The DataSource object supplies your application with connections for accessing the database.

Configuration

General Properties

* **Scope**
cells:was7host01Cell01:nodes:was7host01Node01:servers:server1

* **Provider**
DB2 Universal JDBC Driver Provider (XA)

* **Name**
DB2 Universal JDBC Driver XA DataSource

JNDI name
jdbc/tradedds

Use this data source in container managed persistence (CMP)

Description
DB2 Universal Driver Datasource

After you have defined and saved a data source, you can click the Test Connection button to ensure that the parameters in the data source definition are correct.

A Test Connection operation can have three possible outcomes:

The test can complete successfully, meaning that a connection is successfully obtained to the database using the configured data source parameters.

The test can complete successfully with warnings. This means that while a connection is successfully obtained to the database, warnings were issued. (View the JVM Logs for more details)

The test can fail. A connection to the database with the configured parameters is not obtained (Again, view the JVM Logs for more details)

Data source screen (2 of 3)

Data store helper class name

Select a data store helper class

Data store helper classes provided by WebSphere Application Server

DB2 Universal data store helper
(com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper)

DB2 for iSeries data store helper
(com.ibm.websphere.rsadapter.DB2AS400DataStoreHelper)

Specify a user-defined data store helper

Enter a package-qualified data store helper class name

Security settings

Select the authentication values for this resource.

Authentication alias for XA recovery
was7host01Node01Cell/samples ▼

Component-managed authentication alias
was7host01Node01Cell/samples ▼

Mapping-configuration alias
(none) ▼

Container-managed authentication alias
(none) ▼

Introduction to database connection problems

© 2011 IBM Corporation

The Data store helper class name specifies the name of the DataStoreHelper implementation class that extends the capabilities of your selected JDBC driver implementation class to perform database-specific functions. Be sure to select the one required by your database configuration, otherwise, your application might not work correctly.

The Authentication alias for XA recovery field is used to specify the authentication alias that should be used during XA recovery processing. The database identity for the XA recovery authentication alias on a data source must have authorization to do XA recovery.

The Container-managed authentication alias specifies authentication data, which is a JAAS - J2C authentication data entry, for container-managed sign-on to the resource.

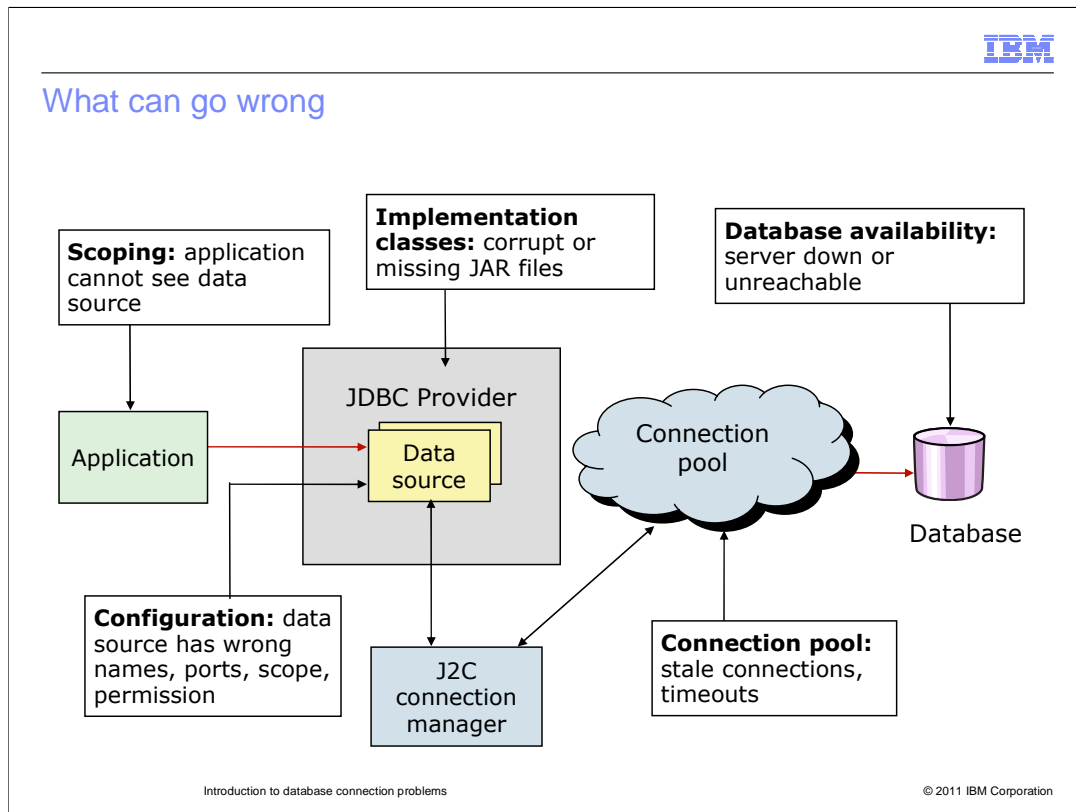
Data source screen (3 of 3)

Common and required data source properties

Name	Value
* Driver type	4 ▾
* Database name	Trade
* Server name	DB2Server
* Port number	50000

The properties shown here are specific to the data source that corresponds to your selected JDBC provider. They are either required by the data source, or are especially useful for the data source. You can find a complete list of the properties required for all supported JDBC providers in the information center.

What can go wrong



Some reasons why your database is having connection problems might include incorrect configuration values, networking problems, stale connections in the connection pool, and database system unavailability.

Creating a database connection: common problems

- Typical problems that arise when configuring a database connection are:
 - Spelling or typographical errors when filling in the parameter values
 - Client JAR files necessary to the JDBC provider are not present
 - Lack of user permissions on the database server
 - Connectivity problems because of network topology or a database that has not been started
- The problems encountered when configuring a database connection are typically easy to fix if you know what to look for and where to look:
 - WebSphere Application Server V7.0 provides immediate feedback when testing a connection.
 - The `systemErr.log` contains explicit information about the actual exception

When configuring a database connection, misspelling any of the database connection properties, or violating any of the case sensitivity rules of a parameter can result in failure. Incorrectly creating a related WebSphere variable or neglecting to make the necessary JAR or native libraries available to the connection will also lead to failure. Inaccurate user authentication information, either in WebSphere Application Server or on the database server, will result in a failed connection.

Before testing a database connection, ensure that the target database has been started and is accessible. The application server SystemOut and SystemErr logs contain failure information should the connection test fail.

Using the test connection service (1 of 2)

- Once the JDBC Provider and the data source have been configured, navigate to **JDBC Providers > data_source_name > Data sources** and click the **Test connection** button.
- The **Test connection** button is also available on the actual data source configuration panel of the administrative console

The screenshot shows the 'JDBC providers' administrative console. The breadcrumb path is 'JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources'. Below the breadcrumb, there is a descriptive paragraph and a 'Preferences' section. In the 'Preferences' section, there are four buttons: 'New', 'Delete', 'Test connection', and 'Manage state...'. The 'Test connection' button is circled in red. Below the buttons is a table of resources.

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	DB2 Universal JDBC Driver XA DataSource	jdbc/tradedts	Node=was7host01Node01,Server=server1	DB2 Universal JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

Total 1

Once the JDBC Provider and the data source have been configured, WebSphere Application Server provides a test connection service for validating data source configurations by way of the administrative console. The Test Connection operation instantiates the data source configuration, gets a connection, and then immediately closes the connection, and returns a result.

Using the test connection service (2 of 2)

- If the data source is configured properly, you will receive a message similar to this example:

```
Messages
  i Test connection for data source DB2UniversalDS on server server1 at node
    rallanNode01 was successful.
```

- If a failure message occurs, then it is time to track down the cause of the failure and resolve it.
- A failure can be caused for several reasons, and is the subject of the next discussion.

If the data source is configured properly, you will receive a “Test connection for data source.....was successful”.

If a failure message occurs, then it is time to track down the cause of the failure and resolve it. Often the failure messages are specific enough to lead to a quick and simple resolution. However, some of the messages report on a side effect of the issue rather than the root cause. In those cases, deeper investigation is necessary in order to track down the root cause of the issue.

JDBC provider configuration problems: other considerations

- If this is a deployment manager scenario, verify that the variables are defined in terms of other variables, such as WAS_INSTALL_ROOT for the application server install path.
- **Important:** Make sure that the scope of each variables you are defining is visible to the variables used in the JDBC provider. Whether they are cell, node, or server scope can be significant.

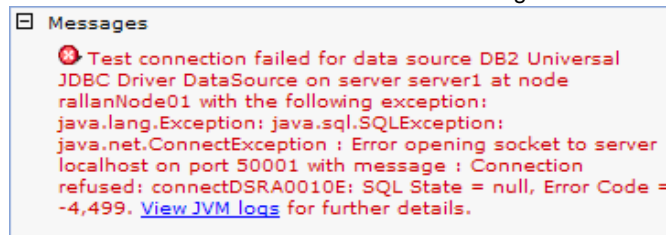
Defining the WebSphere variables in terms of other application server variables allows each node to resolve the variable in the context of its particular environment. Therefore, JAR and native libraries required by the JDBC provider can reside locally, on each node, in a directory relative to the installation path of each node in the cell. The scope in which the variables and the JDBC configuration definitions are defined are extremely important to the visibility of each component.

Defining the driver implementation at the server level will lead to connectivity failures if you attempt to use the data source at the node level.

It is important to make sure that the scope of each variables you are defining is visible to the variables used in the JDBC provider. Whether they are cell, node, or server scope can be significant.

Data source database parameter problems: identification

- The data source is the connection between the application server and the database.
- Common configuration problems consist of:
 - Incorrectly specified database server parameters
 - Incorrect user authentication credentials
- Use the **test connection** service and examine the error message:



```
Messages
* Test connection failed for data source DB2 Universal
JDBC Driver DataSource on server server1 at node
rallanNode01 with the following exception:
java.lang.Exception: java.sql.SQLException:
java.net.ConnectException : Error opening socket to server
localhost on port 50001 with message : Connection
refused: connectDSRA0010E: SQL State = null, Error Code =
-4,499. View JVM logs for further details.
```

- As suggested by this error message, the JVM logs are a good place to begin the problem determination activity

Always keep in mind that the data source is the connection between the application server and the database. The SystemErr log holds the complete stack trace of the exceptions that occur when a data source connection test fails.

Common configuration problems are incorrectly specified database server parameter and incorrect user authentication credentials.

Using the test connection service in the administrative console has a limit on how much of that particular log you can view. As suggested by this error message, the JVM logs are a good place to begin the problem determination activity.

Data source database parameter problems: diagnosis and resolution

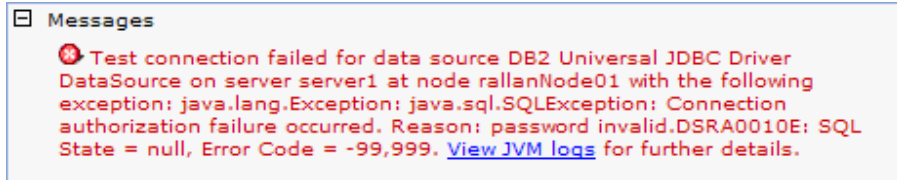
```
3/8/09 11:13:22:458 MST] 0000000a SystemErr      R java.net.SocketException:  
Operation timed out: connect:could be due to invalid address  at  
java.net.PlainSocketImpl.socketConnect(Native Method)
```

- The SystemErr.log tells you that the address provided was not valid for the connection.
- A connectivity problem can be as a result of an incorrect database name, server name, or port number.
 - If this is a large organization, the database administrator might have to be consulted to verify that database name.
 - You can verify that the server name is correct by attempting a ping of the machine or a Telnet session.
 - The port number can have been changed to something other than the default.
 - If it is a DB2 database, the port number can be discovered through the DB2 Configuration Assistant or by exporting the instance parameters by using the DB2 Control Center

Examining the SystemErr log can tell you that the address provided was not valid for the connection. Through your database administrator, verify the database name, server name, and port number. Do not depend upon default values being used. Server name verification can be accomplished by attempting to ping the target machine or through a Telnet session. If this is a large organization, the database administrator might have to be consulted to verify that database name.

Data source database authentication problems: identification

- If you run the **test connection** service and receive an error message like:



- It is signifying a problem with the authentication credentials. This can be a result of:
 - Misspelled user name or password
 - Assigning the wrong JCA credential definition to the data source
 - Missing user account for this user on the database server machine

Before WebSphere Application Server Version V5, data source configuration also included the user name and password information. When Java EE Connector Architecture (JCA) was introduced, the same set of credentials can be mapped to any number of data sources through the data source configuration panel. This is one of the advantages of using a JCA compliant authentication scheme: You only need to fix the one set of credentials and all the data sources will pick up the correction. Regardless of the data source type, a misspelled username or password will cause failures during the test operation. The best course of action for authentication problems is to immediately verify all security settings and assure they are correct.

Data source database authentication problems: diagnosis and resolution

- The exception in the SystemErr.log file says that the password is not valid:

```
[3/23/09 10:40:16:492 CST] 0000003e SystemErr R java.lang.Exception:  
java.sql.SQLException: Connection authorization failure occurred. Reason: password  
invalid.DSRA0010E: SQL State = null, Error Code = -99,999
```

- In this particular case, it was as a result of assigning the incorrect JCA authentication definition to the data source.
 - Inspect all parameters associated with the user authentication and verify that they are correct.
- Coordinate with the database administrator or server security administrator.
 - Verify that the user account actually exists and is a member of the group that is able to access the database

If you happen to be experiencing data source database authentication problems, have a look in the SystemErr.log. In there you see an password not valid exception. From there inspect all parameters associated with the user authentication and verify that they are correct. Also, consult with the database administrator or server security administrator to verify that the user account actually exists and is a member of the group that is able to access the database.

Overview of the DB2 Universal Driver trace facility

- Many JDBC drivers have a trace or debug facility available.

- The DB2 Universal Driver has extensive trace capabilities that can be activated from the WebSphere Application Server V7.0 administrative console.
 - In addition to setting the trace level of the driver, the output file for the trace information can also be specified.
 - The amount of information that is logged depends on the trace level. A value of `-1` will log everything available to the trace file.

The use of tracing might be necessary to resolve your connection problem. Most commercial database drivers will have a trace or debug facility available. For example, the DB2 Universal Driver has extensive trace capabilities which can be activated from the WebSphere Application Server V7 administrative console. The trace output destination and level of tracing can be specified there.

Enabling the DB2 Universal Driver trace facility

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources > DB2 Universal JDBC Driver XA DataSource > Custom properties

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Preferences

New Delete

Select	Name	Value	Description	Required
<input type="checkbox"/>	description		The description of this datasource.	false
<input type="checkbox"/>	traceLevel	-1	Specifies the level of trace, determined by a bitwise combination of constants: TRACE_NONE=0, TRACE_CONNECTION_CALLS=1, TRACE_STATEMENT_CALLS=2, TRACE_RESULT_SET_CALLS=4, TRACE_DRIVER_CONFIGURATION=16, TRACE_CONNECTS=32, TRACE_DROA_FLOWS=64, TRACE_RESULT_SET_META_DATA=128, TRACE_PARAMETER_META_DATA=256, TRACE_DIAGNOSTICS=512, TRACE_SQLJ=1024, TRACE_XA_CALLS=2048, TRACE_META_CALLS=8192, TRACE_DATASOURCE_CALLS=16384, TRACE_LARGE_OBJECT_CALLS=32768, TRACE_SYSTEM_MONITOR=131072, TRACE_TRACEPOINTS=262144, TRACE_ALL=-1.	false
<input type="checkbox"/>	traceFile	DB2Trace.log	The trace file to store the trace output. If you specify the trace file,	false

Introduction to database connection problems © 2011 IBM Corporation

DB2 Universal Driver trace facility in the administrative console where you can enable a specified level of trace. As seen here, the value -1 will log everything available to the trace file. The traceFile parameter is the name of the trace file that is created.



DB2 Universal Driver trace facility output example

```
[ibm][db2][jcc][Time:1143567094964][Thread:WebContainer :
0][DB2ConnectionPoolDataSource@59247092] getPooledConnection (UserName, <escaped>) called
[ibm][db2][jcc] BEGIN TRACE_DRIVER_CONFIGURATION
[ibm][db2][jcc] Driver: IBM DB2 JDBC Universal Driver Architecture 2.4.19
[ibm][db2][jcc] Compatible JRE versions: { 1.3, 1.4 }
...
[ibm][db2][jcc] END TRACE_DRIVER_CONFIGURATION
[ibm][db2][jcc] BEGIN TRACE_CONNECTS
[ibm][db2][jcc] Attempting connection to localhost:50001/ENTDBx
[ibm][db2][jcc] Using properties: { cliSchema=null, clientProgramId=null,...
currentPackagePath=null, portNumber=50001, serverName=localhost,... traceFile=db2trace.log,
useCachedCursor=true, dataSourceName=null, fullyMaterializeLobData=true,
databaseName=ENTDBx, kerberosServerPrincipal=null, jdbcCollection=NULLID, clientUser=null,
traceLevel=-1, currentRefreshAge=-
...
[ibm][db2][jcc] END TRACE_CONNECTS
...
[ibm][db2][jcc] BEGIN TRACE_DIAGNOSTICS
[ibm][db2][jcc][SQLException@65877092] java.sql.SQLException
[ibm][db2][jcc][SQLException@65877092] SQL state = 08004
[ibm][db2][jcc][SQLException@65877092] Error code = -4499
[ibm][db2][jcc][SQLException@65877092] Message = The application server rejected
establishment of the connection. An attempt was made to access a database, ENTDBx, which
was not found.
[ibm][db2][jcc][SQLException@65877092] Stack trace follows
com.ibm.db2.jcc.a.DisconnectException: The application server rejected establishment of the
connection. An attempt was made to access a database, ENTDBx, which was not found.
    at com.ibm.db2.jcc.c.cb.u(cb.java:1595)
```

Introduction to database connection problems

© 2011 IBM Corporation

The information logged by the trace facility is extensive and includes information such as time stamps, thread information, and driver architecture and version information. The trace file information in this example shows the driver information and the exception that is logged as a result of an improperly configured database name in the data source properties. It was generated as a result of executing the test connection operation.

Enabling Oracle JDBC driver tracing

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver \(XA\)](#) > [Data sources](#) > ...

Use this page to specify custom properties that your enterprise requires. Some data sources require additional custom properties for data sources that access external resources.

Preferences

New Delete

Select	Name	Value
You can administer the following resources:		
<input type="checkbox"/>	driverType	thin
<input type="checkbox"/>	oracleLogFileSizeLimit	0
<input type="checkbox"/>	oracleLogFileCount	1
<input type="checkbox"/>	oracleLogFileName	
<input type="checkbox"/>	oracleLogTraceLevel	FINEST
<input type="checkbox"/>	oracleLogFormat	SimpleFormat
<input type="checkbox"/>	oracleLogPackageName	oracle.jdbc.driver

This slide shows an example of enabling Oracle JDBC Driver tracing. It is currently is set to FINEST.

Unit summary

Now that you have completed this unit, you should be able to:

- Test data source connections to databases
- Describe Java EE Connector (J2C) authentication issues
- Describe Java Database Connectivity (JDBC) driver issues
- Use the built-in tester console to test connections
- Enable tracing

Now that this unit has been completed, you should be able to describe and understand issues that can occur with JDBC driver and J2C authentication, and be able to enable the proper tracing and use the built-in tester console to test database connections.



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 email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WA5721G09_DbConProbs_edited.PPT

This module is also available in PDF format at: [../WA5721G09_DbConProbs_edited.pdf](..WA5721G09_DbConProbs_edited.pdf)

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, Rational, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.

© 2011 IBM Corporation