



IBM Software Group

# SW5706 Introduction to database connection problems



© 2007 IBM Corporation

This presentation will act as an introduction to database connection problems when using WebSphere Application Server V6.

## Unit objectives

After completing this unit, you should be able to:

- Describe the role of the JDBC provider
- Describe the role of the data source
- Understand some of the common problems encountered when configuring database connections
- Test a database connection using the WebSphere Application Server Version 6 *Test Connection* service
- Troubleshoot some of the common problems encountered when testing database connections

The goal of this presentation is to understand the roles of the JDBC provider and data source in a WebSphere Application Server V6 database connection and verify the correctness of the database connection through the Test Connection service.

## 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
- 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 WebSphere variables reference both Java and native libraries. JDBC type 1 and 2 drivers require that JAR files and native libraries be referenced, while JDBC type 3 and 4 drivers only require JAR files to be referenced. Other custom variables can be defined in WebSphere Application Server in order to accommodate any vendor-specific implementation.

The screenshot displays the 'JDBC provider screen' in the IBM Software Group administrative console. The left sidebar shows a navigation tree with 'Resources > JDBC Providers' selected. The main content area is titled 'JDBC providers > DB2 Universal JDBC Driver Provider'. A message box at the top indicates that changing the implementation class name will affect template-based data source creation. The configuration panel is divided into 'General Properties' and 'Additional Properties'. The 'Name' field is 'DB2 Universal JDBC Driver Provider', the 'Description' is 'Non-XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support only 1-phase commit processing except in the...', and the 'Implementation class name' is 'com.ibm.db2.jcc.DB2ConnectionPoolDataSource'. The 'Class path' field contains several environment variables. The 'Additional Properties' section shows 'Data sources' and 'Data sources (Version 4)'. At the bottom, there are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

The JDBC provider screen can be accessed through the WebSphere Application Server V6 administrative console by navigating to **Resources > JDBC Providers > (JDBC Provider name)**. This is the panel you use to specify the name, description, any JAR file references, any native library references, and the implementation class name for the provider.

Once the JDBC provider has been defined, at least one data source will need to be created that uses the driver implementation to access the vendor database.

## JDBC provider configuration problems: 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 the variables you are defining are visible to the variables used in the JDBC Provider. Whether they are Cell, Node, or Server scope could be significant.

Defining the database driver provider 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 of the cell. The scope in which the variables, JDBC provider and data source are defined is extremely important to the visibility of each component. Defining the driver implementation at the server level will lead to connectivity failure if you attempt to use the data source at the node level. These driver level variables are not visible at the node scope and will result in class loader exceptions because the system cannot locate the necessary JAR or native libraries. The Test Connection service is a quick, easy way to verify basic connectivity, as it provides immediate feedback when testing a connection, as well as placing explicit information in the SystemErr log in the event of an exception.

## The role of the data source

- A data source provides connectivity to a relational database and is associated with a JDBC Provider.
- A data source is analogous to the J2EE Connector Architecture (JCA) connection factory.
- Configuring a data source consists of supplying database information such as:
  - ▶ Server name
  - ▶ Listening port of the database server
  - ▶ Authentication credentials
  - ▶ Type of driver used for the connection

The data source in WebSphere Application Server V6 represents the actual connection to the database. The data source uses the JDBC provider implementation to connect to the database system. The data source 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 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 (top)

JDBC providers

JDBC providers > Samples Cloudscape JDBC Provider (XA) > Data sources > PLANTSDB

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Configuration

Test connection

**General Properties**

Scope  
cells:rallanCell01:nodes:rallanNode01:servers:server1

Name  
PLANTSDB

JNDI name  
jdbc/PlantsByWebSphereDataSource

Use this Data Source in container managed persistence (CMP)

Description  
Data source for the Plants by WebSphere entity beans

Category

**Data store helper class name**

Select a data store helper class

Data store helper classes provided by WebSphere Application Server

Cloudscape data store helper  
(com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper)

Specify a user-defined data store helper

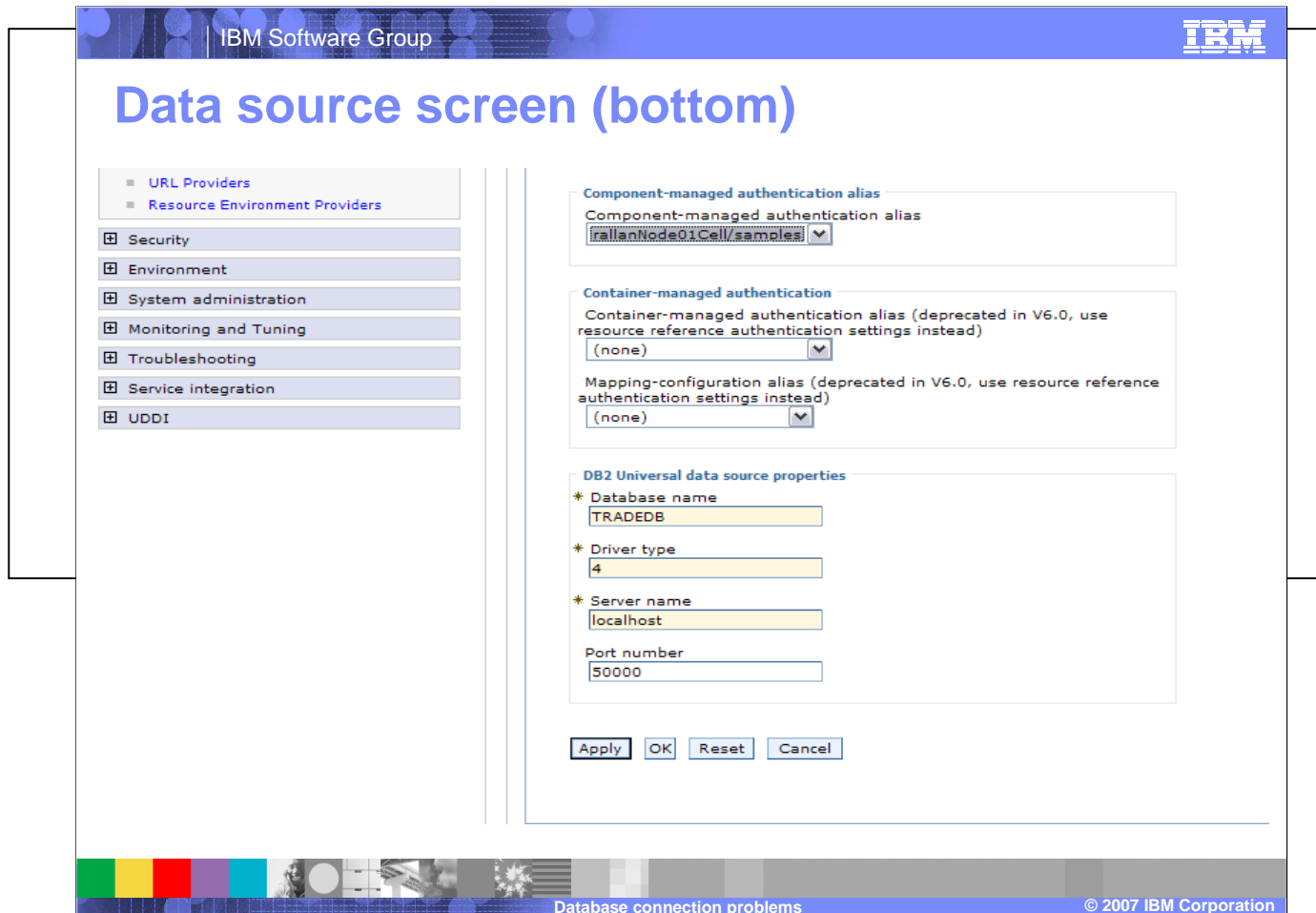
**Additional Properties**

- Connection pool properties
- WebSphere Application Server data source properties
- Custom properties

**Related Items**

- J2EE Connector Architecture (J2C) authentication data entries

The Data source screen in the administrative console contains parameter fields for configuring the database connection. At the top of the screen is the *Test connection* button for testing the connection after configuration is complete. The parameter fields include the scope of the data source, the name of the data source, and the JNDI name of the data source which is used by the application to reference the data source. There is also a check box for specifying whether to use this data source in container managed persistence, and a description, category, and widgets for describing a data store helper class.



Fields at the bottom of the data source definition screen include the *Component-managed authentication alias*, used to associate the JCA authentication properties with the data source, and two deprecated *Container-managed authentication* widgets. Because the data source depicted in this image is for a DB2 Universal database, there are text fields for supplying the *Database name*, the *Driver type*, a *Server name*, and the *Port number*.



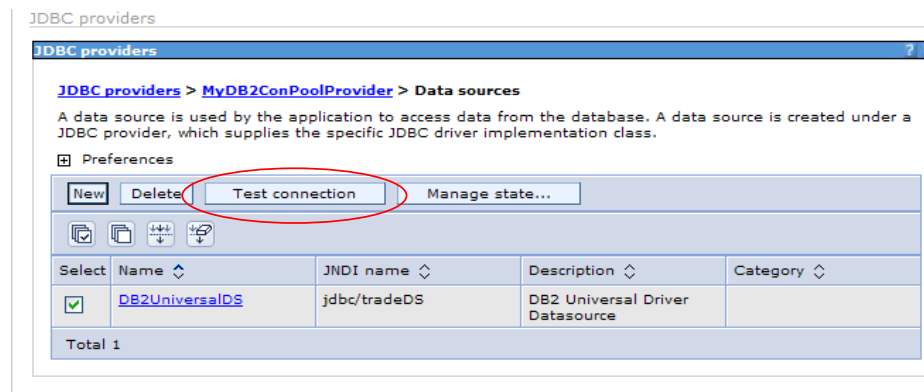
## Creating a database connection: common problems

- Typical problems that arise when configuring a database connection are:
  - ▶ Spelling errors or typos 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 often easy to fix if you know what to look for and where to look:
  - ▶ WebSphere Application Server V6 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 will 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. Ensure that the target database has been started and is accessible.

## Using the Test Connection service (1 of 2)

- Once the JDBC Provider and the data source have been configured, navigate to **JDBC Providers -> (Provider 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



Once the JDBC Provider and the data source have been configured, the *Test connection* button is accessible from two different paths. Once the JDBC Provider and the data source have been configured, the *Test connection* button is accessible from two different paths:

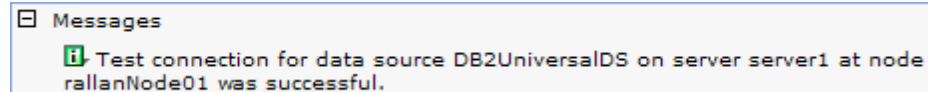
*Resources > JDBC Providers > (Provider Name) > Data sources*

or in the subsequent specific Data Source panel.

The *Test Connection* service uses a default general query of the form `SELECT 1 FROM TABLE1`. This default query string is located in the administrative console panel under *JDBC Providers > (Provider Name) > Data sources > (Data source name) > WebSphere Application Server data source properties* in the *Pretest SQL string* text field. This default query can be changed to any appropriate SQL query. Keep in mind that the data source cannot be tested until any updated values have been applied and saved to the application server configuration files.

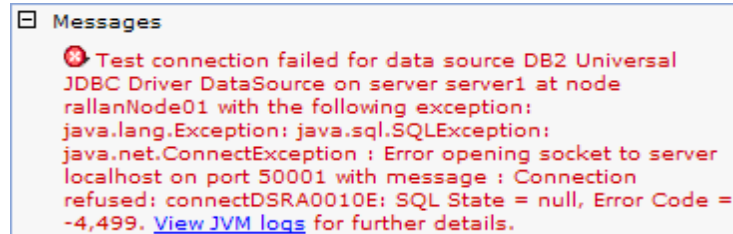
## Using the Test Connection service (2 of 2)

- If the data source is configured properly, you will receive a positive response



Messages  
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

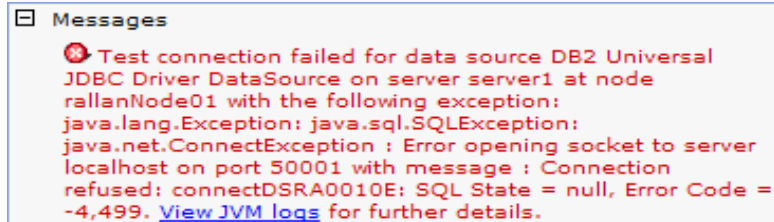


Messages  
Test connection failed for data source DB2 Universal JDBC Driver DataSource on server server1 at node rallanNode01 with the following exception:  
java.lang.Exception: java.sql.SQLException:  
java.net.ConnectException : Error opening socket to server localhost on port 50001 with message :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

WebSphere Application Server V6 provides feedback to the administrative console as soon as the *Test Connection* service completes. The failure messages are often 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.

## Data source database parameter problems: Identification

- The data source is the connection between the application server and the database.
- 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. The administrative console, however, has a limit on how much of that particular log you can view. Therefore, when you use the console to view the log, the last exception to occur might not be available for inspection. In that case, it is advisable to view the log directly on the actual WebSphere Application Server host machine.

## Data source database parameter problems: Diagnosis and resolution

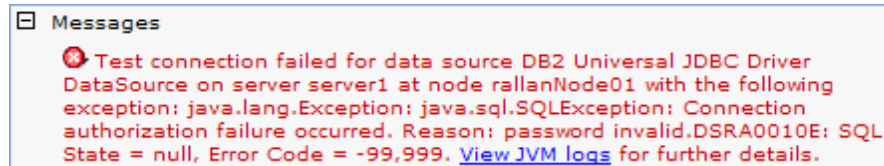
```
3/8/06 11:13:22:458 MST] 0000000a SystemErr      R java.net.SocketException:  
Operation timed out: connect:could be due to invalid address      at  
java.net.PlainSocketImpl.socketConnect(Native Method)
```

- The SystemErr.log tells you that the address provided was invalid for the connection.
- A connectivity problem could be as a result of an incorrect database name, server name, or port number.
  - ▶ The database administrator might have to be consulted to verify that database name.
  - ▶ Verify the server name by attempting a ping of the machine or a Telnet session.
  - ▶ The port number could have been changed to something other than the default
    - If it is a DB2 database, the port number may be discovered through the DB2 Configuration Assistant or by exporting the instance parameters by using the DB2 Control Center.

Examining the SystemErr log in this instance tells you that the address provided was invalid 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.

## Data source database authentication problems: Identification

- If you run the **Test Connection** service and receive an error message like:



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: Connection authorization failure occurred. Reason: password invalid.DSRA0010E: SQL State = null, Error Code = -99,999. [View JVM logs](#) for further details.

- It is signifying a problem with the authentication credentials. This could 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

Authentication exceptions are a common problem, and are often easily remedied. Often a misspelled user name or password could be the culprit, or an incorrect JCA credential may have been assigned to the data source. On the database server machine, a missing account for this user could cause such an exception.

## Data source database authentication problems: Diagnosis and Resolution

- The exception in the SystemErr.log file says that the password is invalid:

```
[3/23/06 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
```

- However, in this particular case, it was as a result of assigning the incorrect JCA authentication definition to the data source. It is important to inspect all of the parameters associated with user authentication and verify that they are correct.
- This might include coordination 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.

The exception found in the SystemErr log for the preceding problem said that the password was invalid. However, in this particular case, the problem was a result of assigning the incorrect JCA authentication definition to the data source. This example underscores the importance of verifying all of the parameters associated with user authentication.

Depending on the database, security services could be provided by the local operating system, LDAP server, or some other form of access control. In a larger organization, verifying user credentials could involve several people and cross over several groups. As was discussed previously, a simple Telnet test to the database server might be necessary in order to verify that the credentials are actually correct and have the necessary authorities required for the data source.

## Overview of the DB2 Universal Driver Trace facility

- Any industry strength database driver will have a trace or debug facility available.
- The DB2 Universal Driver has extensive trace capabilities that can be activated from the WebSphere Application Server V6 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 may 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 V6 administrative console. The trace output destination and level of tracing can be specified there.



## Enabling the DB2 Universal Driver Trace facility

**JDBC providers**

[JDBC providers](#) > [DB2 Universal JDBC Driver Provider](#) > [Data sources](#) > [DB2 Universal JDBC Driver DataSource](#) > [Custom properties](#)

Custom properties that may be required for resource providers and resource factories. 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"/>	<a href="#">description</a>		The description of this datasource.	false
<input type="checkbox"/>	<a href="#">traceLevel</a>	-1	The DB2 trace level for logging to the logWriter or trace file. Possible trace levels are: TRACE NONE = 0, TRACE CONNECTION CALLS = 1, TRACE STATEMENT CALLS = 2, TRACE RESULT SET CALLS = 4, TRACE DRIVER CONFIGURATION = 16, TRACE CONNECTS = 32, TRACE DRDA FLOWS = 64, TRACE RESULT SET META DATA = 128, TRACE PARAMETER META DATA = 256, TRACE DIAGNOSTICS = 512, TRACE SQLJ = 1024, TRACE ALL = -1, .	false
<input type="checkbox"/>	<a href="#">traceFile</a>	db2trace.log	The trace file to store the trace output. If you specify the trace file, the DB2 Jcc trace will be logged in this trace file. If this property is not specified and the WAS database trace group is enabled, then both WebSphere trace and DB2 trace will be logged into the WebSphere trace file.	false

Database connection problems © 2007 IBM Corporation

The DB2 Universal Driver Trace facility is located on the administrative console. The DB2 Universal Driver Trace facility is located on the administrative console at *Resources > JDBC Providers > DB2 Universal JDBC Driver Provider > Data sources > (Data source name) > Custom properties*. The *Custom properties* panel for a given data source will be populated with parameters that are specific to the vendor's driver implementation. For JDBC providers and data sources supported by WebSphere Application Server V6, custom properties will be available. However, for some user defined database connections, such as MySQL, no custom properties are automatically populated in this panel. In this case, you will need to refer to the database driver documentation in order to activate driver tracing.

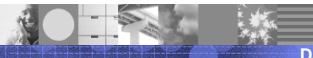
The two parameters of interest are:

**traceLevel** – In this example, it has been set to -1, full trace.

**traceFile** – In this example, the trace file is called db2trace.log and, by default, is written to this file under the WAS\_INSTALL\_ROOT/(application server) directory because of the scope of the JDBC provider. Any relative path can be specified along with the name.

## DB2 Universal Driver Trace facility output

```
[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=NULLLID, 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)
```



The information logged by the trace facility is extensive and includes: time stamps, thread information, driver architecture and version information, environmental information, and verbose low level buffer information for the query and the query results. 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 a SELECT 1 FROM TABLE1 query using the Test Connection service.

## 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 SW5706G08\\_DbConProbs.ppt](mailto:iea@us.ibm.com?subject= Feedback about SW5706G08_DbConProbs.ppt)

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            IBM            WebSphere

J2EE, JDBC, JRE, JVM, 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.

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 (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.

© Copyright International Business Machines Corporation 2007. 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.

