Chapter 11 Extension

WebSphere V6 Network Deployment: HTTP Session Persistence using **DB2 Type-2 Driver**

In the printed version of the book, we gave step by step instructions on how to configure database session persistence using the DB2 Type-4 database driver. Even though configuring and maintaining a Type 2 driver takes more time, you still may want to use a Type 2 driver for various architectural reasons, political reasons or performance reasons (a Type 2 driver may perform slightly better than a Type 4 driver). Figure 11-40 depicts the architecture of the database from a HTTP session persistence viewpoint using a Type-2 DB2 driver. When you use a Type-2 driver you need to install DB2 client software on each WebSphere node (was-host1 and was-host2) where a cluster member (server1 and server2) exists. Some of the steps are similar for both Type-2 and Type-4 drivers. You will be asked to refer to the instructions given in chapter-11 in printed version of the book for all the common or similar tasks. So we suggest you read instructions provided for configuring Type-4 driver in printed version of the book before configuring Type-2 driver.

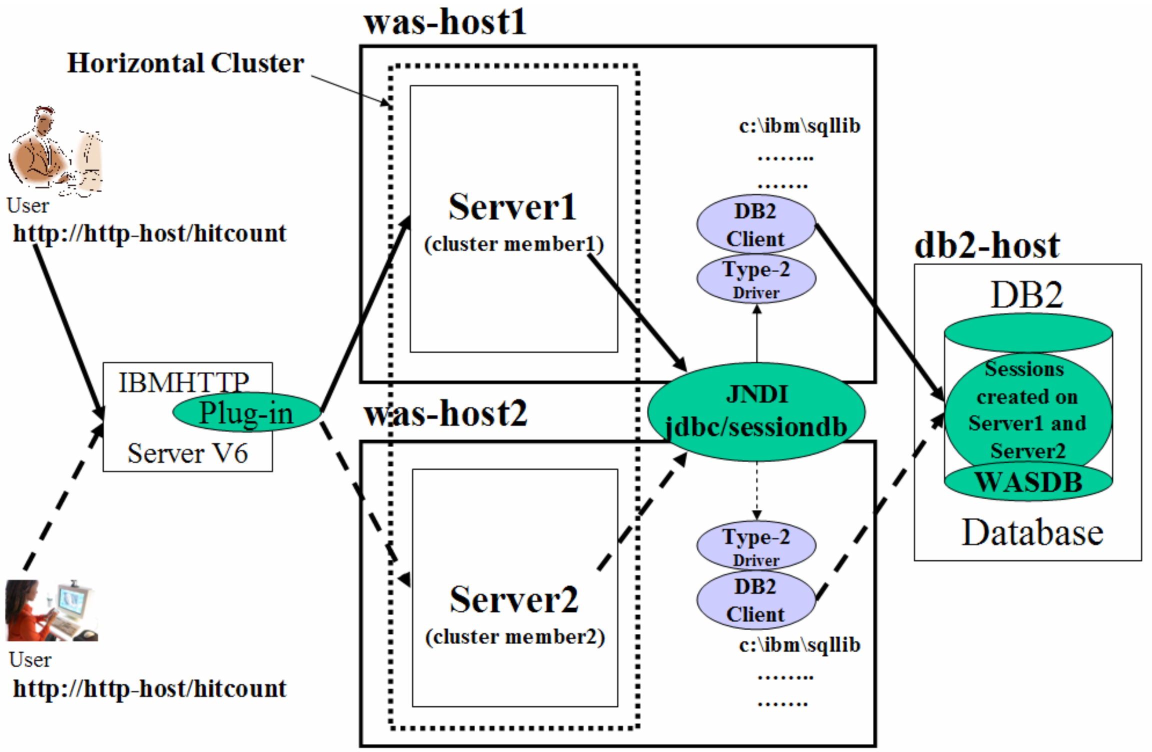


Figure 11-40: Database session persistence architecture using DB2 TYPE-2 Driver

Here are the high-level steps to configure and verify database session persistence using a Type-2 DB2 driver. The process entails the following tasks:

1. Prepare the system for database session persistence.
2. **Install the DB2 client on each WebSphere node (was-host1 and was-host2)**
3. **Configure the DB2 client on each node pointing to the session database (WASDB) on the DB2 server (db2-host).**
4. Create a WebSphere variable pointing to the location of the Type 2 driver files.
5. Create a Java 2 Connector (J2C) authentication entry to access the session database.
6. Create a Java Database Connectivity (JDBC) provider and a data source to access the remote session database through DB2 client.
7. Test the database connection from the admin console.
8. Attach each cluster member to the data source's Java Naming and Directory Interface (JNDI) name to persist sessions.
9. Verify session failover using the hitcount application.

Before configuring database session persistence, make sure you've configured a cluster (vertical or horizontal) and verified that the plug-in for workload management is working. (Refer to Chapter 10 for instructions on setting up and verifying a cluster.).

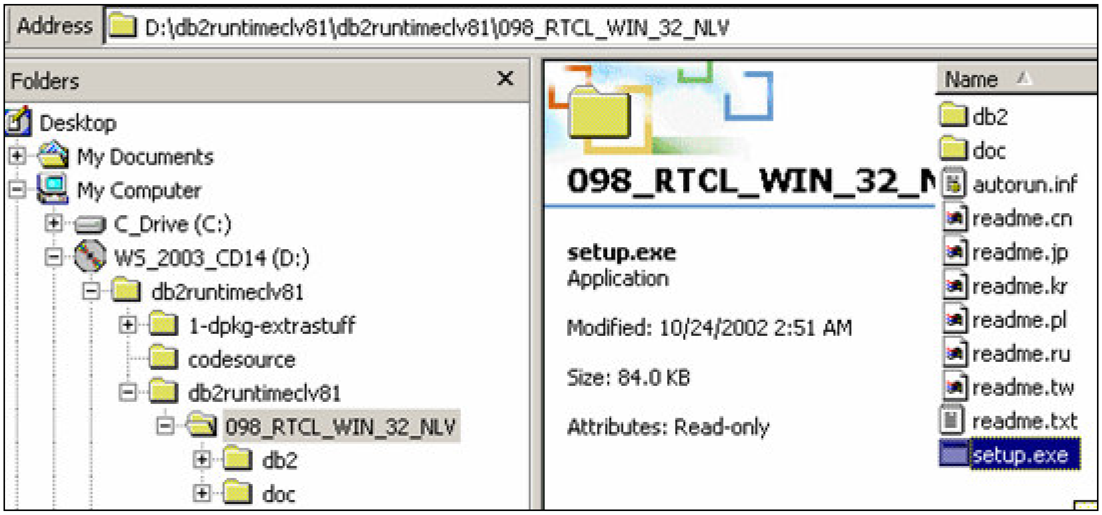
Prepare the System for Database Session Persistence

Step 1. Use the steps under the same task in Chapter-11 in printed version of the book to create, configure the session database (WASDB) for session persistence and prepare the system.

Install the DB2 Client on each WebSphere Node

The instructions given here assume that you have configured horizontal clustering having cluster members on was-host1 and was-host2 as shown in the Figure 11-40. But if your configuration is vertical cluster, then you need to perform the steps on the node where you configured the vertical cluster.

Step 2. Because you're using a DB2 Type 2 driver, you need to install the DB2 client software on each cluster member. Download DB2 Runtime Client 8.x installation zip file (that is same version and fix pack level as DB2 server on db2-host) and unzip to temp directory or mount the CD if it available. Go to the sub directory where the installation program is available. Double click on the installation program. On Windows, start the DB2 client by double clicking on setup.exe (the executable file name depends on the operating system). Make sure that you install the DB2 client on each WebSphere node (was-host1 and was-host2).



Address D:\db2runtimeclv81\db2runtimeclv81\098_RTCL_WIN_32_NLV

setup.exe
Application

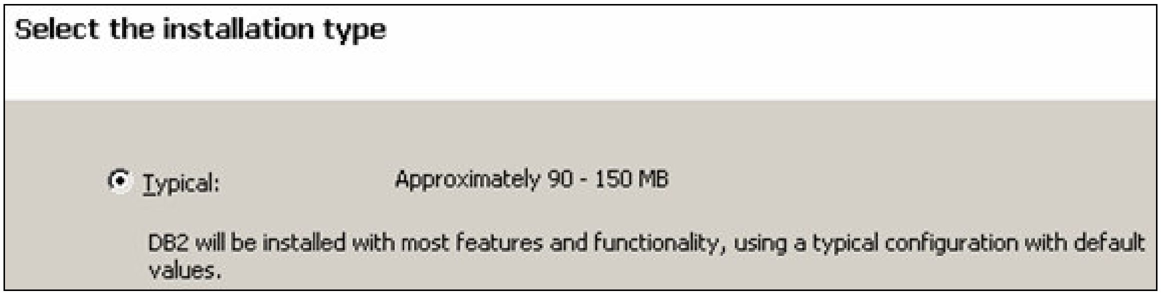
Modified: 10/24/2002 2:51 AM

Size: 84.0 KB

Attributes: Read-only

Figure 11-41: Invoking DB2 client installation program

1. Click **Next** on Welcome screen.
2. Accept the license agreement and click **Next**.
3. Select radio button for **Typical** as the installation type and Click **Next.**



Select the installation type

● Typical: Approximately 90 - 150 MB

DB2 will be installed with most features and functionality, using a typical configuration with default values.

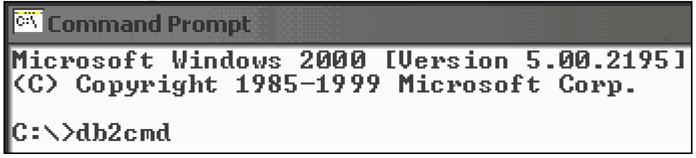
Figure 11-42: Selecting Typical as installation type during DB2 client installation

4. Click on **OK** to ignore the warning message regarding connecting to Remote server.
5. Enter the installation directory for DB2 client as **c:\ibm\sqllib** (<DB2-HOME>). For easy maintenance use the same directory for both DB2 Server and Client installations.
6. Click on **Next** accepting defaults for NetBIOS.
7. On the next screen click on **Install** to start the installation.
8. Click on **Finish** after completion of copying DB2 client files.

```
<begin note>
Make sure that you perform these steps on was-host2 node too.
<end note>
```

Configure the DB2 client on each node pointing to the Remote session database

Step 3. Open a command prompt and type in db2cmd to open the DB2 command window as shown in the screenshot below. Make sure that you perform these steps on each WebSphere node (was-host1 and was-host2).



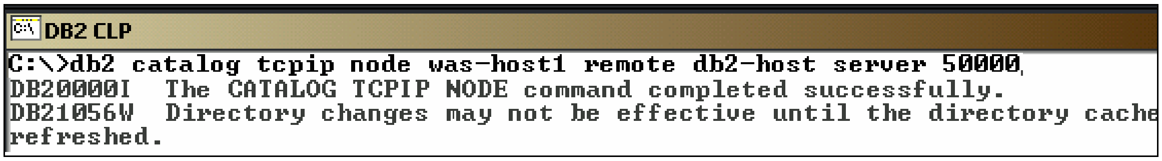
```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\>db2cmd
```

Figure 11-43: Invoking DB2 command prompt

Step 4. Catalog database using the following commands**:**

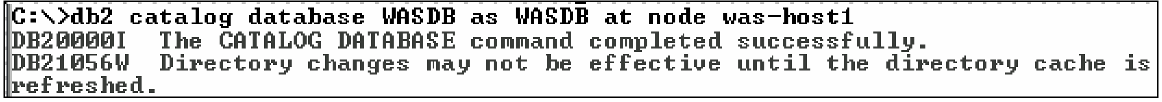
1. 'db2 catalog tcpip node <WAS-HOST> remote <DB2-HOST> server <DB2-PORT>'



```
C:\>db2 catalog tcpip node was-host1 remote db2-host server 50000
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache
refreshed.
```

Figure 11-44: Cataloging the node

2. 'db2 catalog database <SESSIONDB_NAME> as <SESSIONDB-ALIAS> at node <WAS-HOST>. Technically sessiondb-alias name can be different from the sessiondb_name. But we used the same name (WASDB) in our configuration.

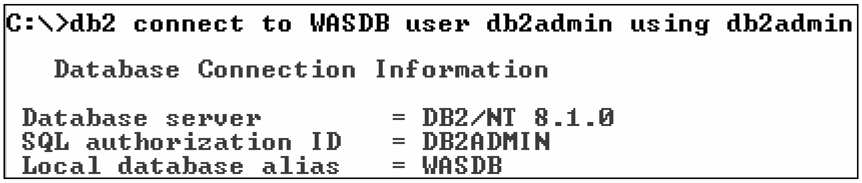


```
C:\>db2 catalog database WASDB as WASDB at node was-host1
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
```

Figure 11-45: Cataloging the database

Step 5. Connect to the remote session database through the sessiondb_alias that you created on the DB2 client using the command below for verification

1. 'db2 connect to <SESSIONDB_ALIAS> user <USERID> using <PASSWORD>'



```
C:\>db2 connect to WASDB user db2admin using db2admin

   Database Connection Information

 Database server        = DB2/NT 8.1.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = WASDB
```

Figure 11-46: Connecting to remote database from DB2 client

<begin note>
Make sure that you perform these steps on was-host2 node too.
<end note>

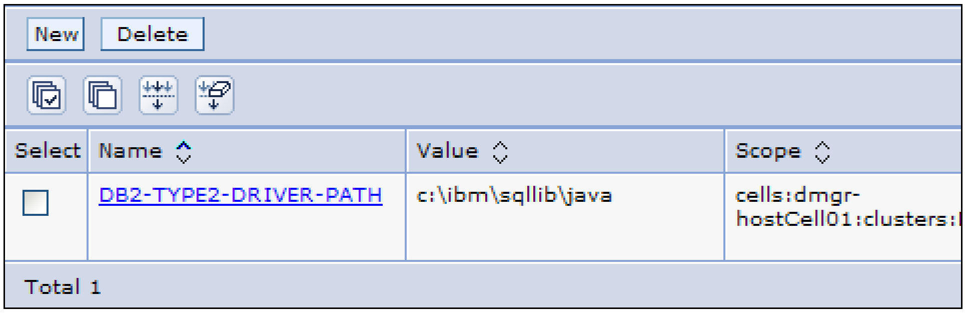
Create a WebSphere Variable Pointing to the Location of the Type 2 Driver

Now that you are able to connect to the database (WASDB) from the command line, it is time to set up a variable in WebSphere that tells the cluster members where the Type 2 driver files are located (<DB2-HOME>\java). You'll create this path variable at the *cluster scope* to make it available to all members of the cluster, even if they're on separate nodes. Use the steps under the same task in Chapter-11 in the printed version of the book to create this WebSphere variable with the following variations.

Step 6. Name the WebSphere variable for DB2 Type-2 driver and define the path where Type 2 driver is available.

a. For the **Name** field, enter DB2-**TYPE2-**DRIVER-PATH (or any name you prefer).
b. For the **Value** field, enter **c:\ibm\sqllib\java** (i.e., <DB2-HOME>\java). Make sure you installed DB2 client to <DB2-HOME> on all nodes of the cluster.
c. Optionally enter a description.
d. Click **OK**, and save the configuration.

After successfully creating the WebSphere variable, you should see its entry in the variables list under cluster scope, as shown in Figure 11-24.



Select	Name	Value	Scope
□	DB2-TYPE2-DRIVER-PATH	c:\ibm\sqllib\java	cells:dmgr-hostCell01:clusters:

Total 1

Figure 11-47: New WebSphere cluster scope variable in variables list

Create a J2C Authentication Entry to Access the Session Database

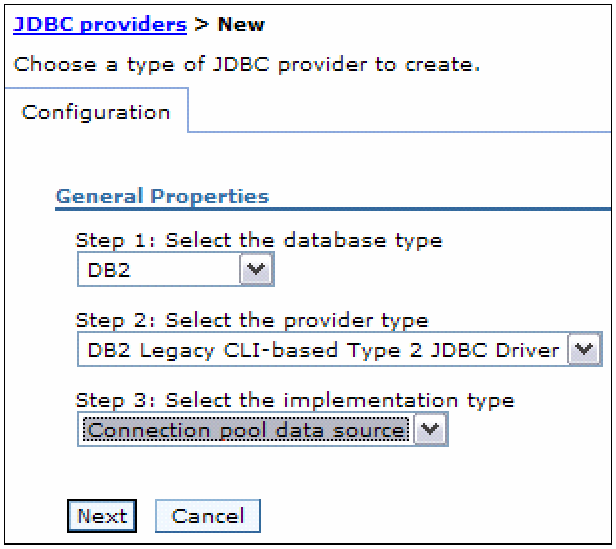
Step 7. Use steps under the same task in Chapter-11 in the printed version of the book to createJ2C Authentication Entry.

Create a JDBC Provider and a Data Source to Access the Session Database

Step 8. Next, create a JDBC provider at the **cluster** scope for each cluster member. To do so, expand **Resources** and select **JDBC Providers**. Click **Browse Clusters**, select the cluster, click **OK**, and click **Apply**. (For a vertical cluster, select VerticalCluster. For a horizontal cluster, select HorizontalCluster.) Then click **New** to create the JDBC provider.

Step 9. On the resulting panel (Figure 11-26), specify the type of JDBC provider for DB2:

a. For the database type, select **DB2**.
b. For the provider type, select **DB2 Legacy JDBC Driver**.
c. For the implementation type, select **Connection pool data source**. (This implementation type is suitable for one-phase commit transactions, especially if performance is your main concern and your operations don't require two-phase commit. For two-phase commit transactions, you'd choose **XA data source**.)
d. Click **Next**.



JDBC providers > New

Choose a type of JDBC provider to create.

General Properties

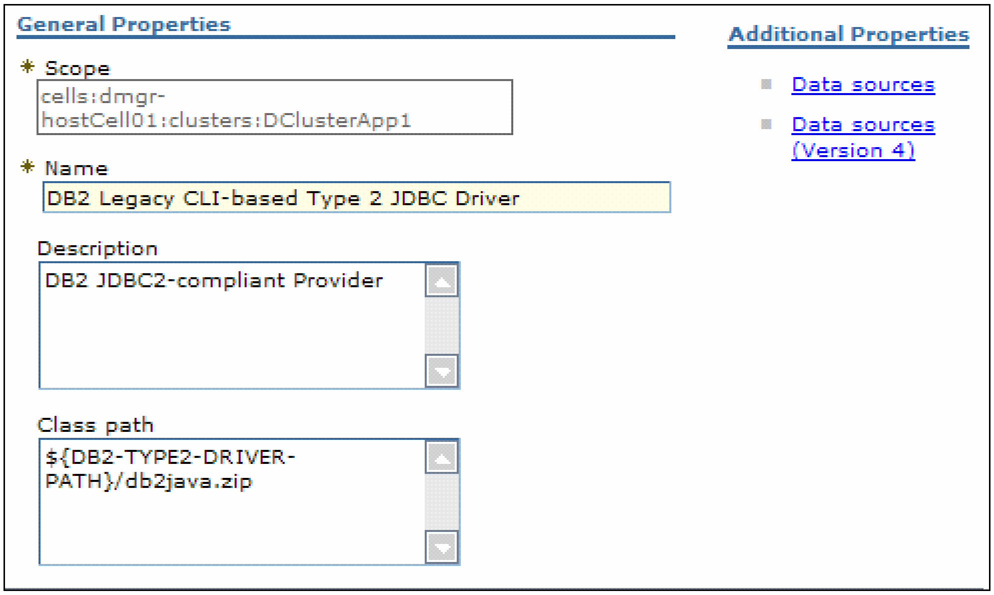
Step 1: Select the database type
DB2

Step 2: Select the provider type
DB2 Legacy CLI-based Type 2 JDBC Driver

Step 3: Select the implementation type
Connection pool data source

Figure 11-48: Specifying the JDBC provider type

Step 10. On the next panel, go to the **Class path** text box, and change the path to "DB2-**TYPE2**-DRIVER-PATH" as shown in the figure. This is the variable you created to point to the directory where you installed DB2 client. (Notice that the variable name contains no underscore characters in our example.) Then scroll down, and click **Apply**.



General Properties

* Scope
cells:dmgr-hostCell01:clusters:DClusterApp1

* Name
DB2 Legacy CLI-based Type 2 JDBC Driver

Description
DB2 JDBC2-compliant Provider

Class path
${DB2-TYPE2-DRIVER-PATH}/db2java.zip

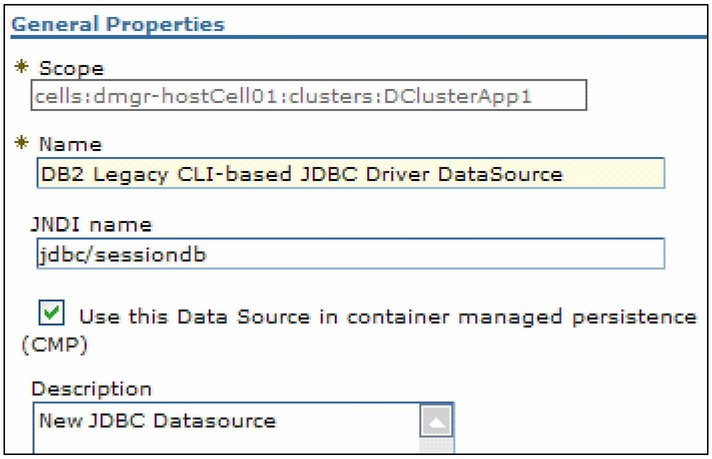
Additional Properties

- Data sources
- Data sources (Version 4)

Figure 11-49: Modifying the JDBC driver path name variable

Step 12. Now, click the **Data sources** link in the panel's upper-right hand corner (under **Additional Properties**), and click **New** on the resulting panel to create a DB2 data source.

Step 13. Enter a JNDI name for the data source that points to the remote session database (WASDB) on the DB2 server (db2-host) through DB2 client configuration. For this example, we'll use the name jdbc/sessiondb..



General Properties

* Scope
cells:dmgr-hostCell01:clusters:DClusterApp1

* Name
DB2 Legacy CLI-based JDBC Driver DataSource

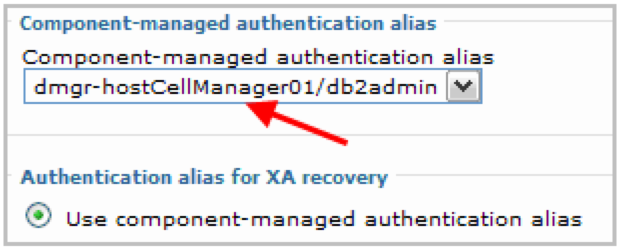
JNDI name
jdbc/sessiondb

☑ Use this Data Source in container managed persistence (CMP)

Description
New JDBC Datasource

Figure 11-50: Specifying a JNDI name for the data source

Step 14. Scroll down until you see the "Component-managed authentication alias" section. From the drop-down list here, select the J2C authentication alias you created earlier.



Component-managed authentication alias

Component-managed authentication alias: dmgr-hostCellManager01/db2admin

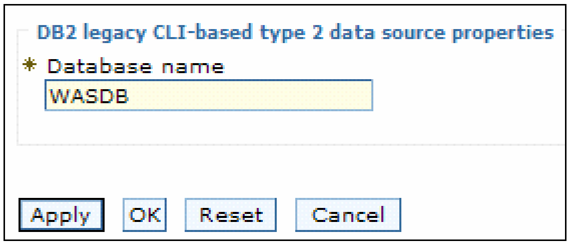
Authentication alias for XA recovery

● Use component-managed authentication alias

Figure 11-51: Specifying the J2C authentication alias

Step 15. Scroll down again to the “DB2 Legacy CLI-based type 2 data source properties” section (Figure 11-31).

a. For the database name, enter the name of the session database, **WASDB**.
b. Click **OK**, and save the configuration.



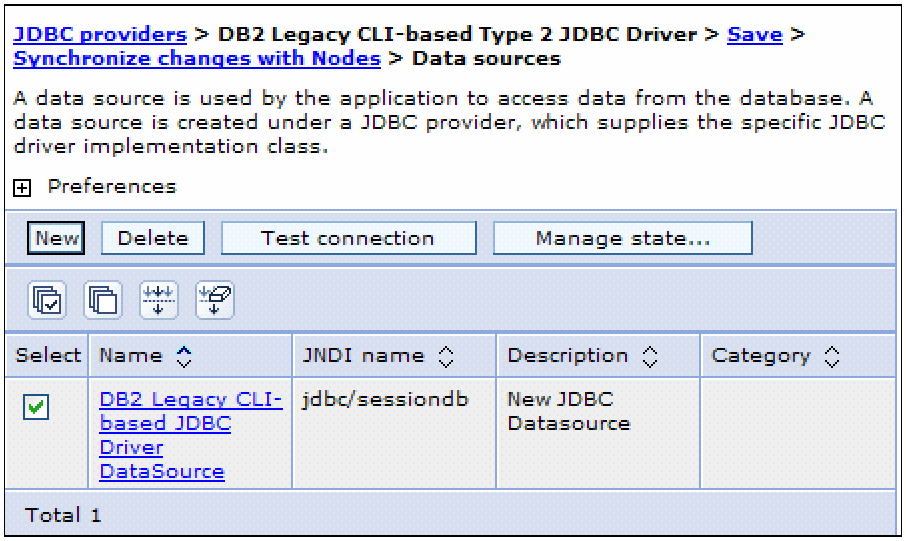
DB2 legacy CLI-based type 2 data source properties

* Database name: WASDB

Figure 11-52: Type 2 data source properties

Test the Database Connection from the Admin Console

Step 7. Use steps under the same task in Chapter-11 in the printed version of the book to test the database connection.



JDBC providers > DB2 Legacy CLI-based Type 2 JDBC Driver > Save > Synchronize changes with Nodes > Data sources

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.

Select	Name	JNDI name	Description	Category
☑	DB2 Legacy CLI-based JDBC Driver DataSource	jdbc/sessiondb	New JDBC Datasource	

Total 1

Figure 11-53: Testing of the new data source connection

Attach Each Cluster Member to the Data Source JNDI Name to Persist Sessions

Step 7. Use the steps under the same task in Chapter-11 in the printed version of the book to configure each cluster member to the data source JNDI name.

Verify Session Failover Using the Hitcount Application

Step 7. Use the steps under the same task in Chapter-11 in the printed version of the book to verify session failover using the hitcount application.