JDBC Configuration on Oracle WebLogic 10.3 (11g) Application Servers

The following steps can be applied to WebLogic 10.3. These instructions apply to applications that leverage the WebLogic Data Source database connectivity framework, which is the most common (and recommended) method. However, if the application code bypasses the WebLogic framework and connects to a database directly (i.e., it instantiates and manages the JDBC driver and connections on its own), use the instructions provided in JDBC Configuration for Standalone or Other Application Server Platforms.

WebLogic Environment Setup

Find the domain library directory for your WebLogic application. This is where the Parasoft JDBC Driver jar file is copied. For example, the lib folder for the WebLogic Medical records sample application domain is usually installed under [WebLogic install dir]/wlserver_10.3/samples/domains/medrec/lib. If you have created your own domain, it might be something like [WebLogic install dir]/user_projects/domains/your_domain/lib.

Creating a New JDBC Data Source

1. Under the Domain Structure, expand Services, then click Data Sources.

2. Identify the Data Source that is currently used by your application and note a) the JNDI name that it is set up with, b) the driver class name, and c) any values set in the Properties field.

3. Open the Targets tab, uncheck the servers on which the Data Source is deployed, then click Save. Alternatively (or in addition to this action), you might change the JNDI name so that it does not conflict with the same JNDI name that we will be providing to the new JDBC data source.
4. Return to the Data Sources section and do the following:
   1. Click New.
   2. Choose the Generic Data Source option.
   3. Under JDBC Data Source Properties, specify the Name and the JNDI Name from step 2.
   4. Set the database type to Other.
   5. Click Next.

5. Set Database Driver to Other, then click Next.
6. Under Transaction Options, either leave the default option or choose another option that suits your current application and database driver configuration settings. Then, click Next.

7. Under Connection Properties, specify the configuration that pertains to your existing database connection settings (and matches your notes from step 2 above). Then, click Next.
8. Under Test Database Connection, do the following:
   1. Provide the Parasoft JDBC Driver class name that relates to your original JDBC driver. For details, see JDBC Driver Implementation Classes.
   2. Specify the JDBC connection URL prefixed with the Parasoft JDBC Driver string. For example:
   3. Provide any properties that may be needed for your database (as obtained from step 2 above).
   4. Provide the following properties:
      virtualizeGroupId
      virtualizeServerUrl
5. (Optional) Provide a Test Table Name or a SQL query to test the new Data Source, then click Test Configuration. System properties may need to be added if any were defined in the original data source (from step 2).

The messages section at the top will indicate the test result. If it succeeds, you will see the following message:
If there is an error, you will see this message:

![Connection test failed message]

In this case, use the message to resolve the error. Look at the Parasoft JDBC Driver log file at the location you provided in the logFilePath property for more details provided by the Parasoft JDBC Driver. Additional logging details are available from the WebLogic server side in the domain’s log location, typically under the servers/DomainName/logs folder.

For example:

```
[WebLogic install dir]/wlserver_10.3/samples/domains/medrec
/servers/MedRecServer/logs
```

Once the test connection succeeds, click **Next**.

**Note**

It is possible that “Connection test succeeded” will be displayed despite the fact that Virtualize Server is down (when the Parasoft JDBC Driver is in Record, Virtualize, or Hybrid mode) or if something is misconfigured. Make sure to look at the Virtualize Console view for incoming SQL queries in record mode, or refer to the Parasoft JDBC Driver log file to check for the absence of any errors.

9. Under Select Targets, select the server(s) where you want to deploy the new data source (these should be the same as those used by the original data source), then click **Finish**.

---

**Alternative Method: Modifying Application Settings To Use the Parasoft JDBC Driver With a Different JNDI Name**

This approach can be applied when you do not want to modify the original data source—perhaps in cases where the same data source is used by multiple applications and you want to virtualize the database connections for only a specific application. Under these circumstances, you may be able to modify the application’s data source JNDI binding through the administrative console. Deployed applications can be accessed through the **Deployments** section under the Domain Structure tree.

```
Domain Structure
```

```
medrec
+ Environment
  Deployments
```
Locate the desired application module and modify the JNDI name for the JDBC data source that it is using.

**JNDI Naming Tip**

JNDI names often take a hierarchical directory-like form to indicate subcontexts. For example, if your original Data Source JNDI name is `jdbc/zOS_DB2DataSource`, then you should avoid renaming it to `jdbc/zOS_DB2DataSource/Original` (using a `/` divider) because that can cause JNDI binding conflicts in the application. Instead, use names such as `jdbc/zOS_DB2DataSourceOriginal` and avoid your JNDI name being interpreted as a subcontext.

**Alternative Method: Calling the Original Data Source**

You can call the original data source if it would be easier for you to reference the existing data source (e.g., if you have many options set) or if you are using a custom data source factory.

To configure a WebLogic Data Source to call the original data source:

1. Copy the Parasoft JDBC Driver jar file to the domain library directory for your WebLogic application (see WebLogic Environment Setup for details).
2. In the Configuration> Connection pool tab, specify the following:
   - **URL**: Enter the URL using a format such as `datasource://jdbc/db2Live`, where the first part (`datasource://`) sets the referencing and the second part (`jdbc/db2Live`) is the JNDI name of the data source to be referenced.
   - **Driver Class Name**: Enter `com.parasoft.xtest.jdbc.virt.driver.JDBCProxyDriver`.
   - **Properties**: Set `virtualizeServerUrl` and `virtualizeGroupId`.

Note:

- The required properties (`virtualizeGroupId` and `virtualizeServerUrl`) are set on the resource as well. No system properties are needed.
• A recommended best practice is to add virtualizeServerPath to indicate which Virtualize server path the driver will send request information to. Once your asset is created, you will need to deploy it on that same path. This allows you to send data only to this specific asset.