JDBC Configuration on WebSphere Application Servers

The following steps can be applied to application servers versioned from 6.0 and later. These instructions apply to applications that leverage the WebSphere Data Source database connectivity framework, which is the most common and recommend method. However, if the application code bypasses the WebSphere 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.

Prerequisite

To prepare for the configuration, copy the Parasoft JDBC Driver jar to a location on the application server. You will need to provide WebSphere the path to the driver to through the administrative console.

The configuration involves:

1. Creating a New JDBC Provider for the Parasoft JDBC Driver
2. Modifying the Original Data Source Settings
3. Creating a New JDBC Data Source with the Parasoft JDBC Driver or Alternative Method: Modifying Application Settings To Use the Parasoft JDBC Driver With a Different JNDI Name

Creating a New JDBC Provider for the Parasoft JDBC Driver

These steps are intended to create a JDBC provider definition within WebSphere configured with the Parasoft JDBC Driver:

1. Log in to your WebSphere Application Server's Integrated Solutions Console.
2. Open Resources> JDBC and click JDBC Providers.
3. Select Node> Server scope (the 3rd option) then click New.
4. Complete the Create New JDBC Driver screen, then click Next:
4. Database type: Choose User-defined.
5. Implementation class name: Specify one of the values that are available (this depends on the database vendor JDBC driver currently in use). See JDBC Driver Implementation Classes for details.
6. Name: Enter any name you want to assign this Parasoft JDBC Driver.

In the Enter database class path information screen, enter the path to the location to where you placed the Parasoft JDBC Driver jar file.

5. In the Enter database class path information screen, enter the path to the location to where you placed the Parasoft JDBC Driver jar file.

6. Complete the Summary screen by clicking Finish.

Modifying the Original Data Source Settings

This section explains how to adjust the WebSphere Data Source configuration that is currently being used by the application whose database connection you wish to virtualize.

You will modify the JNDI name of the data source so that you can define a separate new Data Source with that same JNDI name. This will prevent naming conflicts with the existing data source. It also saves you from having to change the application itself (or its deployment configuration) within WebSphere. If more than one deployed application uses that same Data Source definition and you do not want to impact these other applications, see Alternative Method: Modifying Application Settings To Use the Parasoft JDBC Driver With a Different JNDI Name, which explains how to alter the application settings with a different/new Data Source configuration (instead of altering the existing Data Source configuration).

To prepare for the setting adjustment:

1. Go to the Data Source that is currently being used by your application.
2. Note the settings and configurations in the main configuration page. You will need the property names and values later when you are configuring a new Data Source for the Parasoft JDBC Driver.
3. Click Custom Properties.
4. Note any property names and values there. You will need them later when you are configuring a new Data Source for the Parasoft JDBC Driver.
5. Note the JNDI name for the data source. The JNDI name is what the application uses in order to look up the JDBC data source to use. You will use that same value in the new Data Source for the Parasoft JDBC Driver.

To perform the adjustment (using the PlantsByWebSphere sample application [provided with WebSphere] as an example):

1. Change the JNDI name to a different value, then click Apply. For instance, if the original value was `jndi/PlantsByWebSphereDataSource`, it could be changed to `jndi/PlantsByWebSphereDataSourceOriginal`.

Creating a New JDBC Data Source with the Parasoft JDBC Driver

Next, you will use WebSphere's New Data Source wizard to create a new Data Source configuration that uses the JDBC Provider you created.

1. Open Resources> JDBC> Data Sources.
2. Complete the **Enter basic data source information** screen as follows:
   - **Data source name**: Enter any name you want to assign to this data source.
   - **JNDI name**: Enter the JNDI name you collected for the application's data source (see Modifying the Original Data Source Settings). For example, in the case of the PlantsByWebSphere sample application, you would use jndi/PlantsByWebSphereDataSource.

3. Complete the **Select JDBC provider** screen as follows:
   - a. Choose **Select an existing JDBC provider**.
   - b. Select the provider that you created in Creating a New JDBC Provider for the Parasoft JDBC Driver.

4. In the **Enter database specific properties for the data source** screen, leave the default Data store helper class name. If the application which uses the data source uses CMP, then also select **Use this data source in container managed persistence (CMP)**.
5. In the **Specify security aliases** screen, leave the defaults, then click **Next**.

6. Click **Finish**.

7. In the new Parasoft data source that was created, add the same properties you noted in the application's data source configuration (as described in **Modifying the Original Data Source Settings**), plus additional properties for the Parasoft JDBC Driver data source.
   
   - The Parasoft JDBC Driver specific properties that need to be defined are:
     - virtualizeServerUrl
     - virtualizeGroupId
   
   - For example, for the PlantsByWebSphere sample application, you could use the following settings:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtualizeServerUrl</td>
<td><a href="http://devel66i9000">http://devel66i9000</a></td>
<td>false</td>
</tr>
<tr>
<td>virtualizeGroupId</td>
<td>db2</td>
<td>false</td>
</tr>
</tbody>
</table>

8. Save the changes to the WebSphere master configuration.
9. Restart WebSphere.
10. Return to the Parasoft JDBC Data source and click **Test connection**.

If the configuration was performed properly, there should be no errors. 1 or 2 warnings are expected:

This warning is shown because GenericDataStoreHelper is used in the Parasoft JDBC data source.

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

These instructions can be applied in the cases where you do not want to modify the original data source—perhaps in the cases where the same data source is used by multiple applications and you want to virtualize the database connections for only a specific application. These instructions apply to enterprise applications that can have their data source JNDI bindings modified in the WebSphere through the administrative console.

1. Open **Applications> Enterprise Applications** (or **Applications> Application Types> Web-Sphere enterprise applications** in other versions of WebSphere).

2. Click the desired application. This is the application that will be reconfigured to use the new Parasoft JDBC Driver data source.
**For WebSphere 6.1 and earlier**

If the application uses CMP persistence (which in the case with the PlantsByWebSphere example), click “Provide default data source mapping for modules containing 2.x entity beans” and modify the current target resource JNDI name with the JNDI name you provided in Creating a New JDBC Data Source with the Parasoft JDBC Driver (step 2), but prefix it with eis/ and postfix it with _CMP. For example, if you configured the Parasoft JDBC Driver data source with jdbc/ParasoftJDBCDataSource, here you will provide eis/jdbc/ParasoftJDBCDataSource_CMP

After that, click Map data sources for all 2.x CMP beans and modify the current target resource JNDI name with the JNDI name you provided in Creating a New JDBC Data Source with the Parasoft JDBC Driver (step 2) in the previous section, but prefix and postfix with eis/ and _CMP accordingly.

**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.