

Using the DataDirect Connect[®] for JDBC[®] Drivers with JBoss

Introduction

This document explains the steps required to use the DataDirect Connect[®] for JDBC[®] drivers with the JBoss Application Server¹. These steps include:

1. Install DataDirect Connect for JDBC.
2. Create a data source. A sample data source file, `datadirect-ds.xml`, is available with this document.
3. Generate the JavaServer Page (JSP) test page. A sample JSP test page is provided in the `JBossTestWeb.war` file, which is available with this document.
4. Deploy the JBoss web application to the JBoss server.
5. Run the JBoss web application.
6. Map a data source to a local Java Naming and Directory Interface (JNDI) name in a Session Enterprise Java Bean (EJB).
7. Specify the data source for an Entity EJB.

Detailed information about each of these steps is included in the following sections.

1. Install DataDirect Connect[®] for JDBC

1. Install the DataDirect Connect for JDBC drivers using the instructions in the [DataDirect Connect for JDBC Installation Guide](#).
2. Copy the DataDirect Connect for JDBC jar files (`util.jar`, `base.jar`, `db2.jar`, `informix.jar`, `oracle.jar`, `sqlserver.jar`, and `sybase.jar`) from:

`install_dir/lib`

to:

`JBoss_home/server/server_config/lib`

where `install_dir` is your DataDirect Connect for JDBC installation directory, `JBoss_home` is your JBoss installation directory, and `server_config` is your server configuration directory.

¹ The steps in this paper were generated using JBoss 3.2.3. These steps may vary for other versions of JBoss.

For example, to configure the DataDirect Connect *for* JDBC drivers for the default JBoss server configuration, copy the driver jar files to *JBoss_home/server/default/lib*.

2. Create a Data Source

1. In the *JBoss_home/server/server_config/deploy* directory, create a data source file named *datadirect-ds.xml*. The file name must end with the characters *-ds.xml* so that the JBoss server can recognize it as a data source file; however, the file name can be prefixed with any set of characters.
2. Edit the file as shown in the following example, adding or modifying the required XML tags as described in this step. The example shown is contained in the file *datadirect-ds.xml*, which is available from the same location you obtained this document.

- The value of the `<jndi-name>` tag specifies the JNDI name, which is used to look up the data source. In the following example, the JNDI name used to look up the data source is *ds/TestDS*. JBoss maps the driver to the global space. An application can look up the data source using the string:

```
java:ds/TestDS
```

- Modify the value of the `<driver-class>` tag to specify the appropriate class name for the driver to use. For example, the following code specifies the class name of the DataDirect Connect *for* JDBC SQL Server driver.

```
<driver-class>
  com.ddtek.jdbc.sqlserver.SQLServerDriver
</driver-class>
```

- Modify the value of the `<connection-url>` tag to specify the correct connection information for the driver and database server to use. For example, the following code specifies connection information used by the DataDirect Connect *for* JDBC SQL Server driver to connect to the server named *myserver* on port 1433.

```
<connection-url>
  jdbc:datadirect:sqlserver://myserver:1433
</connection-url>
```

- Modify the value of the `<user-name>` and `<password>` tags to specify a valid user name and password for the database server. For example, the following code specifies the user name *test* and the password *secret*:

```
<user-name>test</user-name>
<password>secret</password>
```

- The `<connection-property>` tag specifies DataDirect Connect *for* JDBC driver-specific connection properties. For example, the following code

sets the value of the DataDirect Connect *for* JDBC SQL Server driver connection property SendStringParametersAsUnicode to false.

```
<connection-property name="sendStringParametersAsUnicode">
    false
</connection-property>
```

- The tags following the <!--pooling parameters--> comment in the following example are properties that control JBoss connection pooling. Refer to the JBoss documentation for details on setting these properties.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ===== -->
<!--                                           -->
<!-- DataDirect Data Sources                 -->
<!--                                           -->
<!-- ===== -->

<!--
    See the generic_ds.xml file in the doc/examples/jca folder
    for examples of properties and other tags you can specify
    in data sources
-->

<datasources>

    <!-- JBossTest Data Source -->
    <local-tx-datasource>
        <jndi-name>ds/TestDS</jndi-name>
        <connection-url>
            jdbc:datadirect:sqlserver://myserver:1433
        </connection-url>
        <driver-class>
            com.ddtek.jdbc.sqlserver.SQLServerDriver
        </driver-class>
        <user-name>test</user-name>
        <password>secret</password>

        <!-- Driver Specific Options -->
        <connection-property name="sendStringParametersAsUnicode">
            false
        </connection-property>

        <!--pooling parameters-->
        <min-pool-size>5</min-pool-size>
        <max-pool-size>100</max-pool-size>
        <blocking-timeout-millis>5000</blocking-timeout-millis>
        <idle-timeout-minutes>15</idle-timeout-minutes>
    </local-tx-datasource>

</datasources>
```

3. Generate the JSP Test Page

You must create a JSP page that uses the data source you created in the previous step. A sample JSP page named `JBossTest.jsp` is provided in the `JBossTestWeb.war` file, which is available from the same location you obtained this document. This JSP page includes the following code to look up the data source and obtain a connection to the database from the data source.

```
InitialContext ctxt = new InitialContext();
DataSource ds = (DataSource) ctxt.lookup("java:ds/TestDS");
con = ds.getConnection();
```

4. Deploy the JBossTest Web Application to the JBoss Application Server

Copy the `JBossTestWeb.war` file, available with this document, to the directory `JBoss_home/server/server_config/deploy`.

5. Run the JBossTest Web Application

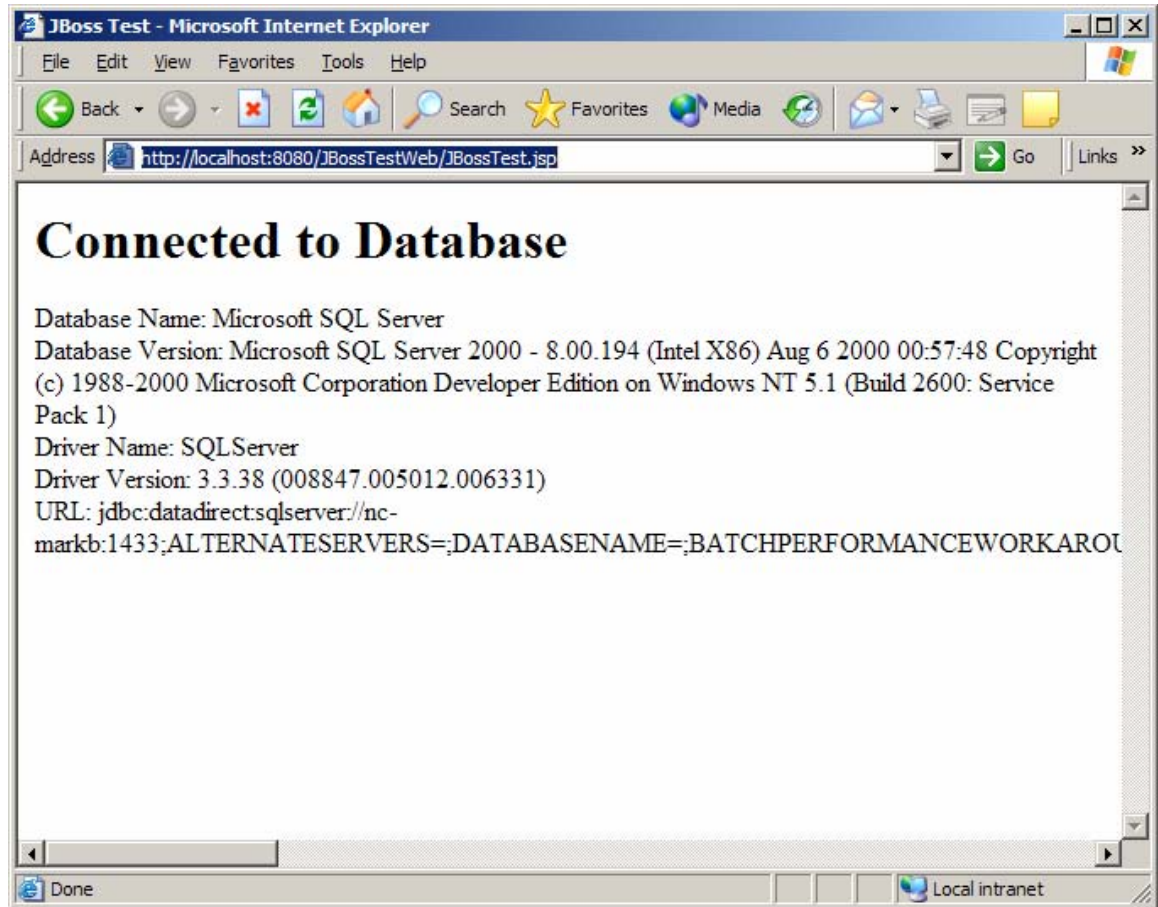
1. Start the JBoss application server by running `run.bat` or `run.sh`, located in the `JBoss_home/bin` directory. To start a configuration other than the default configuration, use the `-c` option when executing the script. For example to run the all configuration, use the following command:

```
run -c all
```

2. Open a web browser and enter the following URL to display the JBossTest web page:

```
http://localhost:8080/JBossTestWeb/JBossTest.jsp
```

The contents of the page display the version information of the driver and the database server to which it connects, if the DataDirect Connect *for* JDBC driver has been installed and configured correctly. For example, the following figure shows version information for the DataDirect Connect *for* JDBC SQL Server driver connecting to a database server running Microsoft SQL Server 2000.



6. Map the Data Source to a Local JNDI Name in a Session EJB

Typically an EJB does not use the global JNDI name to look up the data source. Instead, it uses a logical JNDI name that is mapped to the global JNDI name of the data source.

To map a logical JNDI name to the global JNDI data source name in a Session EJB, declare a resource reference in the JBoss-specific deployment descriptor file `jboss.xml`. A resource reference is defined by adding the tag `<resource-ref>` as a child of the `<session>` tag as shown in the following example.

```
<session>
  <ejb-name>SupportedDatabases</ejb-name>
  <jndi-name>SupportedDatabasesBean</jndi-name>
  <local-jndi-name>SupportedDatabasesLocal</local-jndi-name>
  <resource-ref>
    <res-ref-name>jdbc/TestDS</res-ref-name>
    <jndi-name>java:/ds/TestDS</jndi-name>
  </resource-ref>
</session>
```

The value of the `<res-ref-name>` tag is the logical JNDI name an EJB uses to look up the data source. The value of the `<jndi-name>` tag is the global name of the data source to which the logical name is mapped.

Refer to your JBoss documentation for more information about JBoss-specific deployment descriptors.

7. Specify the Data Source for an Entity EJB

To specify the data source to be used with an Entity EJB, specify the global JNDI name of the data source in the JBoss-specific Container Manager Persistence (CMP) deployment descriptor, `jbosscmp-jdbc.xml`. The value of the `<datasource>` tag specifies the global JNDI name of the data source to use with Entity EJBs. The `<datasource>` tag can be specified as a child of the `<defaults>` tag as shown in the following example, or it can be specified as a child of a particular `<entity>` tag.

```
<defaults>
  <datasource>java:/ds/TestDS</datasource>
  <datasource-mapping>MS_SQLSERVER</datasource-mapping>
</defaults>
```

Refer to your JBoss documentation for more information about JBoss-specific deployment descriptors.

We welcome your feedback! Please send any comments concerning documentation, including suggestions for other topics that you would like to see, to:

docgroup@datadirect.com

FOR MORE INFORMATION

800-876-3101

Worldwide Sales

Belgium (French).....	0800 12 045
Belgium (Dutch).....	0800 12 046
France	0800 911 454
Germany	0800 181 78 76
Japan	0120.20.9613
Netherlands	0800 022 0524
United Kingdom	0800 169 19 07
United States	800 876 3101

Copyright © 2005 DataDirect Technologies Corp. All rights reserved. DataDirect Connect is a registered trademark of DataDirect Technologies Corp. in the United States and other countries. Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Other company or product names mentioned herein may be trademarks or registered trademarks of their respective companies.



DataDirect Technologies is focused on data access, enabling software developers at both packaged software vendors and in corporate IT departments to create better applications faster. DataDirect Technologies offers the most comprehensive, proven line of data connectivity components available anywhere. Developers worldwide depend on DataDirect Technologies to connect their applications to an unparalleled range of data sources using standards-based interfaces such as ODBC, JDBC and ADO.NET, as well as cutting-edge XML query technologies. More than 250 leading independent software vendors and thousands of enterprises rely on DataDirect Technologies to simplify and streamline data connectivity. DataDirect Technologies is an operating company of Progress Software Corporation (Nasdaq: PRGS).

www.datadirect.com