Quick Start: Progress® DataDirect Connect® Series for JDBC™

The following basic information allows you to connect with and test your Progress® DataDirect Connect® Series for JDBC™ drivers immediately after installation. For installation instructions, refer to the DataDirect Connect for JDBC Series Installation Guide.

Connecting to a Database

Once the DataDirect Connect for JDBC drivers are installed, you can connect from your application to your database in two ways: using the JDBC Driver Manager or using a JNDI data source. The following sections provide information you need to get started using the drivers with the Driver Manager. For information on using data sources, refer to the DataDirect Connect Series for JDBC User’s Guide.

Click the following links for specific information about each driver:

DataDirect Connect® for JDBC
- **DB2 Driver**
- **Informix Driver**
- **MySQL Driver**
- **Oracle Driver**
- **PostgreSQL Driver**
- **Progress OpenEdge Driver**
- **SQL Server Driver**
- **Sybase Driver**

DataDirect Connect XE® for JDBC
- **The Driver for Apache Hive**
- **Greenplum Driver**
- **Salesforce Driver**

DB2 Driver

JVM Requirement

The DB2 driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

```
install_dir/lib/db2.jar
```

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a Java Virtual Machine (JVM), you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.
The classname and connection URL syntax for the DB2 driver are:

<table>
<thead>
<tr>
<th>Driver Classname:</th>
<th>com.ddtek.jdbc.db2.DB2Driver</th>
</tr>
</thead>
</table>
| Connection URL:  | **DB2 for Linux/UNIX/Windows:**
|                  | jdbc:datadirect:db2://servername:port;DatabaseName=db_name
|                  | [;property=value[;...]] |
|                  | **DB2 for z/OS and i platforms:**
|                  | jdbc:datadirect:db2://servername:port;
|                  | LocationName=location_name[;property=value[;...]] |

For example, to connect to a DB2 for Linux/UNIX/Windows database named ACCTNG on the server named corpserver1 on port 50000, the following URL would be used:

jdbc:datadirect:db2://corpserver1:50000;DatabaseName=ACCTNG

Similarly, to connect to a DB2 for z/OS or i platform database with a location name of ACCTNG on the server named corpserver1 on port 50000, the following URL would be used:

jdbc:datadirect:db2://corpserver1:50000;LocationName=ACCTNG

Refer to the *DataDirect Connect Series for JDBC User's Guide* for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

**Examples**

**DB2 for Linux/UNIX/Windows:**

```java
// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.db2.DB2Driver");

// Establish the Connection
String url = "jdbc:datadirect:db2://corpserver1:50000;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());
```

**DB2 UDB for z/OS and i Platforms:**

```java
// Register the driver with the driver manager
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.db2.DB2Driver");

// Establish the Connection
String url = "jdbc:datadirect:db2://corpserver1:50000;LocationName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");
```
Informix Driver

JVM Requirement

The Informix driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/informix.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Informix driver are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>com.ddtek.jdbc.informix.InformixDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL</td>
<td>jdbc:datadirect:informix://servername:port;InformixServer=server_name[;property=value;...]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG in the Informix server instance named corpSrvr on the server named corpserver1 on port 2003, the following URL would be used:

jdbc:datadirect:informix://corpserver1:2003;InformixServer=corpSrvr;DatabaseName=ACCTNG

Refer to the **DataDirect Connect Series for JDBC User's Guide** for a complete description of the connection URL syntax and a description of the connection properties that can be specified.

Example

```java
// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.informix.InformixDriver");
```
// Establish the Connection
String url = 
"jdbc:datadirect:informix://corpserver1:2003;InformixServer=corpSrvr;
DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());

MySQL Driver

JVM Requirement

The MySQL driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/mysql.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class
name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection()
method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip
this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the MySQL driver are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>com.ddtek.jdbc.mysql.MySQLDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL</td>
<td>jdbc:datadirect:mysql://hostname:[port] [;property=value[;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 3306,
the following URL would be used:

jdbc:datadirect:mysql://corpserver1:3306;DatabaseName=ACCTNG

Refer to the DataDirect Connect Series for JDBC User's Guide for a complete description of the
connection URL syntax and descriptions of the connection properties that can be specified.
Example

// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.mysql.MySQLDriver");

// Establish the Connection
String url = "jdbc:datadirect:mysql://corpserver1:3306;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());

Oracle Driver

JVM Requirement

The Oracle driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/oracle.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Oracle driver are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>com.ddtek.jdbc.oracle.OracleDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL</td>
<td>jdbc:datadirect:oracle://servername:port;[;property=value [;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to Oracle instance named test on the server named corpserver1 on port 1521, the following URL would be used:

jdbc:datadirect:oracle://corpserver1:1521;ServiceName=test

Refer to the DataDirect Connect Series for JDBC User's Guide for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.
Example

// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.oracle.OracleDriver");

// Establish the Connection
String url = "jdbc:datadirect:oracle://corpserver1:1521;ServiceName=test";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: "+metaData.getDatabaseProductName());
System.out.println("Database Version: "+metaData.getDatabaseProductVersion());

PostgreSQL Driver

JVM Requirement

The PostgreSQL driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/postgresql.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class
name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip
this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the PostgreSQL driver are:

<table>
<thead>
<tr>
<th>Driver Classname:</th>
<th>com.ddtek.jdbc.postgresql.PostgreSQLDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL:</td>
<td>jdbc:datadirect:postgresql://hostname:[port] [:property=value[;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 5432,
the following URL would be used:

jdbc:datadirect:postgresql://corpserver1:5432;DatabaseName=ACCTNG

Refer to the DataDirect Connect Series for JDBC User's Guide for a complete description of the
connection URL syntax and descriptions of the connection properties that can be specified.
Example

// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.postgresql.PostgreSQLDriver");

// Establish the Connection
String url =
"jdbc:datadirect:postgresql://corpserver1:5432;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());

Progress OpenEdge® Driver

JVM Requirement

The OpenEdge driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/openedgewp.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class
name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection()
method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip
this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Progress OpenEdge driver are:

<table>
<thead>
<tr>
<th>Driver Classname:</th>
<th>com.ddtek.jdbc.openedge.OpenEdgeDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL:</td>
<td>jdbc:datadirect:openedge://servername:port[;property=value [;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 2003,
the following URL would be used:

jdbc:datadirect:sqlserver://corpserver1:2003;DatabaseName=ACCTNG
Refer to the *DataDirect Connect Series for JDBC User’s Guide* for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

**Example**

```java
// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.openedge.OpenEdgeDriver");

// Establish the Connection
String url =
   "jdbc:datadirect:openedge://corpserver1:2003;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
   metaData.getDatabaseProductVersion());
```

**SQL Server Driver**

**JVM Requirement**

The SQL Server driver requires a Java SE 5 or higher JVM.

**Environment Setup**

Set your system CLASSPATH to include the following entry:

`install_dir/lib/sqlserver.jar`

**Driver Manager**

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the `DriverManager.getConnection()` method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The class name and connection URL syntax for the SQL Server driver are:

<table>
<thead>
<tr>
<th>Driver Classname:</th>
<th>com.ddtek.jdbc.sqlserver.SQLServerDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL:</td>
<td>jdbc:datadirect:sqlserver://servername:port[;property=value [...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 1433, the following URL would be used:

`jdbc:datadirect:sqlserver://corpserver1:1433;DatabaseName=ACCTNG`
Refer to the DataDirect Connect Series for JDBC User’s Guide for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

Example

```
// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.sqlserver.SQLServerDriver");

// Establish the Connection
String url =
   "jdbc:datadirect:sqlserver://corpserver1:1433;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
   metaData.getDatabaseProductVersion());
```

Sybase Driver

JVM Requirement

The Sybase driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

```
install_dir/lib/sybase.jar
```

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Sybase driver are:

<table>
<thead>
<tr>
<th>Driver Classname:</th>
<th>com.ddtek.jdbc.sybase.SybaseDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL:</td>
<td>jdbc:datadirect:sybase://server-name:port[;property=value [;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 4100, the following URL would be used:

```
jdbc:datadirect:sybase://corpserver1:4100;DatabaseName=ACCTNG
```
Refer to the *DataDirect Connect Series for JDBC User’s Guide* for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

**Example**

```java
// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.sybase.SybaseDriver");

// Establish the Connection
String url = "jdbc:datadirect:sybase://corpserver1:4100;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " + metaData.getDatabaseProductVersion());
```

**The Driver for Apache Hive™**

**JVM Requirement**

The driver for Apache Hive requires a Java SE 5 or higher JVM.

**Environment Setup**

Set your system CLASSPATH to include the following entry:

```
install_dir/lib/hive.jar
```

**Driver Manager**

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

**IMPORTANT:** If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Driver for Apache Hive are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>com.ddtek.jdbc.hive.HiveDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL</td>
<td>jdbc:datadirect:hive://servername:port[;property=value[;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 10000, the following URL would be used:

```
jdbc:datadirect:hive://corpserver1:10000;DatabaseName=ACCTNG
```
Refer to the DataDirect Connect Series for JDBC User’s Guide for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

Example

// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.hive.HiveDriver");

// Establish the Connection
String url = "jdbc:datadirect:hive://corpserver1:10000;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());

Greenplum Driver

JVM Requirement

The Greenplum driver requires a Java SE 5 or higher JVM.

Environment Setup

Set your system CLASSPATH to include the following entry:

install_dir/lib/greenplum.jar

Driver Manager

To establish a database connection using the Driver Manager, an application must register the class name of the JDBC driver with the Driver Manager, and then invoke the DriverManager.getConnection() method while supplying a connection URL.

IMPORTANT: If using Java SE 6 or higher as a JVM, you do not need to register the driver and can skip this step. Java SE 6 and higher automatically registers the driver with the JDBC Driver Manager.

The classname and connection URL syntax for the Greenplum driver are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>com.ddtek.jdbc.greenplum.GreenplumDriver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection URL:</td>
<td>jdbc:datadirect:greenplum://hostname:[port]</td>
</tr>
<tr>
<td></td>
<td>[:property=value[;...]]</td>
</tr>
</tbody>
</table>

For example, to connect to a database named ACCTNG on the server named corpserver1 on port 5432, the following URL would be used:

jdbc:datadirect:greenplum://corpserver1:5432;DatabaseName=ACCTNG
Refer to the DataDirect Connect Series for JDBC User's Guide for a complete description of the connection URL syntax and descriptions of the connection properties that can be specified.

Example

// Register the driver with the driver manager.
// If using Java SE 6 or higher, you can omit this step.
// Java SE 6 and higher automatically registers the driver.
Class.forName("com.ddtek.jdbc.greenplum.GreenplumDriver");

// Establish the Connection
String url = "jdbc:datadirect:greenplum://corpserver1:5432;DatabaseName=ACCTNG";
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());

Salesforce Driver

JVM Requirement

The Salesforce driver requires a Java SE 7 or higher JVM to comply with Salesforce security standards.

Environment Setup

Set your system CLASSPATH to include the following entry:

`install_dir/jdbc/lib/sforce.jar`

NOTE: The dependent jar files that are installed in the `install_dir/jdbc/lib` folder with the sforce.jar must remain in the same directory as the sforce.jar file for the driver to be able to find them without explicitly adding them to the class path. For more information about these files, refer to the DataDirect Connect Series for JDBC User's Guide.

Windows Example

`CLASSPATH=.;c:\connect\jdbc\lib\sforce.jar`

UNIX/Linux Example

`CLASSPATH=./home/user1/jdbc/lib/sforce.jar`

Driver Manager

To establish a database connection using the Driver Manager, the application must invoke the DriverManager.getConnection() method supplying a connection URL.
The classname and connection URL syntax for the Salesforce driver are:

<table>
<thead>
<tr>
<th>Driver Classname</th>
<th>Connection URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.ddtek.jdbc.sforce.SForceDriver</td>
<td>jdbc:datadirect:sforce://ServerName</td>
</tr>
</tbody>
</table>

To connect to a production instance of a Salesforce organization, for example, you would use the following URL:

jdbc:datadirect:sforce://login.salesforce.com

NOTE: If you are connecting to a Salesforce instance configured for a security token, you must supply one in the connection URL. You can supply a security token by specifying one with the SecurityToken connection property or by appending the security token to the Password property. For more information about connection URL syntax, SecurityToken, and other connection properties, refer to the DataDirect Connect Series for JDBC User's Guide.

Example

```java
// Establish the Connection
String url = "jdbc:datadirect:sforce://login.salesforce.com;
Connection con = DriverManager.getConnection(url, "test04", "test04");

// Verify the Connection
DatabaseMetaData metaData = con.getMetaData();
System.out.println("Database Name: " + metaData.getDatabaseProductName());
System.out.println("Database Version: " +
    metaData.getDatabaseProductVersion());
```

Tuning Drivers for Optimal Performance

To tune a driver for optimal performance, run the DataDirect Connect for JDBC Performance Tuning Wizard. The Performance Tuning Wizard leads you step-by-step through a series of questions about your application. Based on your answers, the Wizard provides the optimal settings for performance-related connection properties.

The Wizard runs as an applet within a browser window. To start the Wizard, launch the following file from your browser, where `install_dir` is your DataDirect Connect for JDBC installation directory:

`install_dir/wizards/index.html`

NOTE: Security features set in your browser can prevent the Performance Wizard from launching. A security warning message is displayed. Often, the warning message provides instructions for unblocking the Performance Wizard for the current session. To allow the Performance Wizard to launch without encountering a security warning message, the security settings in your browser can be modified. Check with your system administrator before disabling any security features.
Testing the Connection

DataDirect Test™ for JDBC is a JDBC testing tool developed by Progress DataDirect and distributed with DataDirect Connect for JDBC. For more information about DataDirect Test for JDBC, refer to the DataDirect Connect Series for JDBC Reference. Take the following steps to connect and test the connection with DataDirect Test for JDBC:

1. Start DataDirect Test for JDBC by running testforjdbc.bat or testforjdbc.sh in the install_dir/testforjdbc directory, where install_dir is your installation directory.
2. Click Press Here to Continue.
3. From the Connection menu, select Connect to DB.
4. Select the appropriate template for your database, specify server name, port, database name, and any other required properties. Click OK.
5. A new connection window opens. You may proceed with further testing.

© 2016. Progress Software Corporation. All rights reserved.