

# 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

## 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:

<b>Driver Classname:</b>	com.ddtek.jdbc.db2.DB2Driver
<b>Connection URL:</b>	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:

```
// 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:

```
// 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");

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

# Informix Driver

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.informix.InformixDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:informix://servername:port; InformixServer=server_name[;property=value[;...]]</code>

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

```
// 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

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.mysql.MySQLDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:mysql://hostname:[port] [;property=value[;...]]</code>

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

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.oracle.OracleDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:oracle://servername:port;[;property=value [;...]]</code>

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

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.postgresql.PostgreSQLDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:postgresql://hostname:[port] [;property=value[;...]]</code>

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

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.openedge.OpenEdgeDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:openedge://servername:port[:property=value[:...]]</code>

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

```
// 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

## 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 classname and connection URL syntax for the SQL Server driver are:

<b>Driver Classname:</b>	com.ddtek.jdbc.sqlserver.SQLServerDriver
<b>Connection URL:</b>	jdbc:datadirect:sqlserver://servername:port[;property=value [;...]]

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

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.sybase.SybaseDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:sybase://servername:port[;property=value [;...]]</code>

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

```
// 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™

## 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:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.hive.HiveDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:hive://servername:port[;property=value [;...]]</code>

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:hive://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

```
// 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://corpserver1:4100;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());
```

# Greenplum Driver

## 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:

<b>Driver Classname:</b>	com.ddtek.jdbc.greenplum.GreenplumDriver
<b>Connection URL:</b>	jdbc:datadirect:greenplum://hostname:[port] [;property=value[;...]]

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

## 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, your application must register the class name of the JDBC driver with the Driver Manager using the `Class.forName()` method and specifying the driver class name as the argument. The application must then invoke the `DriverManager.getConnection()` method 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 Salesforce driver are:

<b>Driver Classname:</b>	<code>com.ddtek.jdbc.sforce.SForceDriver</code>
<b>Connection URL:</b>	<code>jdbc:datadirect:sforce://<i>ServerName</i></code>

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

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

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.sforce.SForceDriver ");

// 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 Wizard is installed with the product and is also available from the DataDirect Progress web site (<http://www.datadirect.com/products/jdbc/wizard-agreement.html>). 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 that was installed with the product, 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 pure java 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](#). Use the following steps to test your connection:

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, and specify the correct server name, Informix server name (if testing an Informix connection), port, and database name. Click **OK**.
5. A new connection window opens if all of the connection information was entered correctly. If the connection cannot be established, an error is displayed in the JDBC/Database Output window.

© 2013. Progress Software Corporation. All rights reserved.

06/2013