



Building an ODBC Driver with Java

This Sample Driver for Text Files Shows You How

By Simba Technologies Inc.

Prerequisites

This white paper assumes that you are familiar with ODBC and C/C++ development under Microsoft® Visual Studio®, as well as familiarity with Java development.

You will need:

- Microsoft Visual Studio 2008.
- Version 6 of the Java Development Kit: available from <http://java.sun.com/javase/downloads/index.jsp>.
- Version 6 of the Java Runtime Environment: available from <http://java.sun.com/javase/downloads/index.jsp>.
- Either Ant or Eclipse to build the Java packages. Ant can be downloaded from <http://ant.apache.org/>. Eclipse can be downloaded from <http://www.eclipse.org/>.

Introduction

This white paper describes the steps required to build an ODBC driver using Java and SimbaEngine SDK. The resulting driver provides read-only access to the same comma-separated-values (CSV) text files used by the SimbaEngine Quickstart sample C++ driver, and the two drivers are functionally equivalent. The sample code demonstrates one way to use Java to easily connect SimbaEngine to Java-coded data sources that might be hard to reach using C++ alone.

There are more and more Java-coded data sources storing critical data. In addition, while many Java installations may not look like data sources, they can be thought of that way because they provide an important view of the underlying data to applications. Take, for example, the business model layer of a model-view-controller system. While it might seem obvious to provide users who want to query business data with access to the underlying database, they really expect to see the data in the terms dictated by the business model. In this case, it makes more sense to query the data through the business model. Since many of these installations are coded in Java, you need to access the model using Java. However, applications like Microsoft Excel® and SAP® Business Objects Crystal Reports® access data using ODBC. An ODBC driver built

using Java and SimbaEngine SDK provides all the components required to connect ODBC applications to a Java-coded data source.

SimbaEngine SDK provides a C/C++ ODBC interface and SQL engine that require only a data source code layer, the Data Record Manager or DRM, to connect to a data source. The DRM code maps the data source API to the SQL engine. DRM code has been written that maps many data types and architectures to the SQL engine. Once a DRM layer is written for a data source, the data becomes available to the wide range of ODBC-enabled applications.

The data source does not have to be relational, or even arranged in tables and columns; although, in our example case it is. The Java driver views each CSV file as a single table. Each row ends with a carriage return, and the first row contains the column names. Like the Quickstart sample C++-built driver, the Java-built driver supports five native types – char, integer, timestamp, float and varchar.

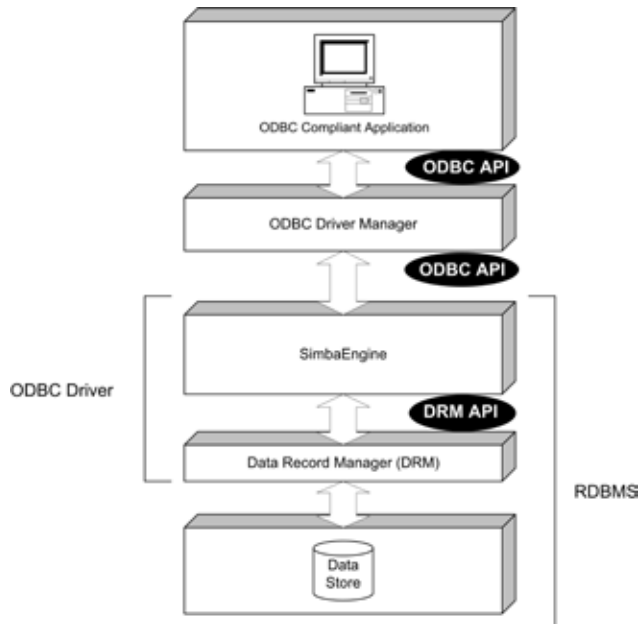


Fig. 1: A typical SimbaEngine ODBC driver architecture using a local data source.

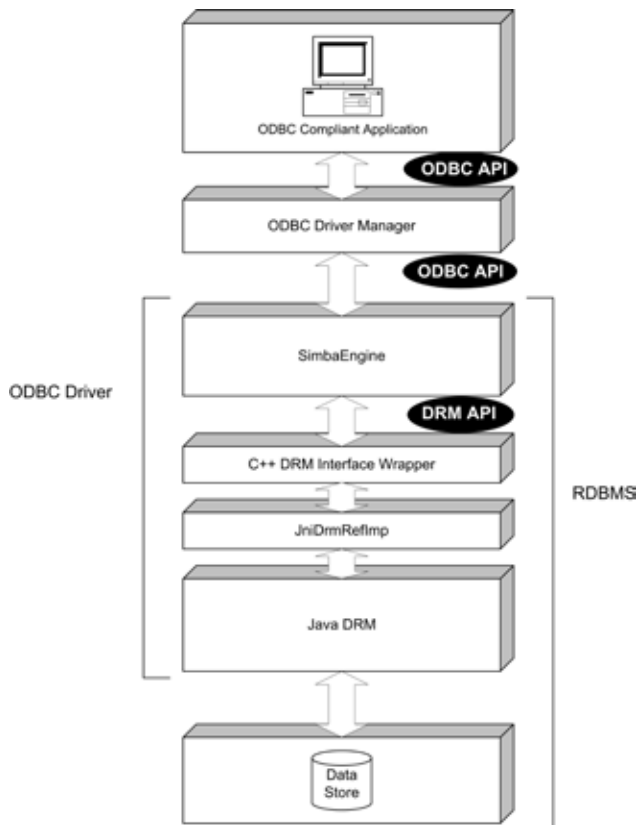


Fig. 2: The SimbaEngine ODBC driver architecture built using Java showing various layers.

The sample files for the Java DRM are in the `<Install Dir>\SourceCode\Engine\DataRecordManager\QuickstartJavaSample` directory installed with SimbaEngine SDK. The Java-built ODBC driver consists of the following two primary components and sub-components.

JniDrmReflmp

This C++ library assumes the role of a traditional DRM layer. It resolves the function calls made by the SQL engine and uses the Java Native Interface (JNI) to load the Java driver. It then acts as a bridge between the SQL engine and the Java DRM layer. More information and the JNI specification can be found at <http://java.sun.com/javase/6/docs/technotes/guides/jni/index.html>.

Java

This is the Java DRM layer itself. It is called by the JNI and provides access to the CSV files. The Java DRM layer is split into several packages located under the `Java\JavaDRM` directory:

- **src\com\simba\drm\interfaces**
This package provides the interfaces that define the Java DRM API. These interfaces must be implemented by the Java DRM layer. The API is based on the C++ DRM API.
- **src\com\simba\drm\interceptor**
src\com\simba\drm\requests
The `DrmProxy` class proxies all requests made by SimbaEngine via the JNI to the Java DRM layer. Request forwarding is handled using the *Command pattern* pattern. The request classes, argument classes and their corresponding factories can be found in the `requests` package.
- **src\com\simba\drm\structs**
This package contains Java counterparts for the C structures used throughout the DRM API. **JniDrmReflmp** has a corresponding set of JNI proxy classes for each structure.

The actual Java sample DRM code is located in the `src\com\simba\samples\quickstart` package. The contents of which are as follows:



- **src\com\simba\samples\quickstart**
This package contains the implementations of key DRM API interfaces – IDriver, ISession, IDatabase and DrmProxy.
- **src\com\simba\samples\quickstart\core**
This package contains utility and exception classes used by the Java-coded driver.
- **src\com\simba\samples\quickstart\table**
This package contains classes for working with both real and virtual tables.
- **src\com\simba\samples\quickstart\types**
This package contains helper classes for dealing with the DRM data types.

Building the Java Quickstart ODBC Driver on Windows

Building the Java Quickstart ODBC driver is easy once you have SimbaEngine SDK and your Java tool properly installed on your computer. Follow these steps to get yourself set up to build a complete driver.

1. The **JniDrmReflmpl** library is built as part of the SimbaEngine SDK. Open SimbaEngine.Net2008.sln in Visual Studio 2008 and build the Debug and Release configurations of the entire solution.
2. The **JavaDrm** component of the driver can be built in two different ways. It can be built outside of an IDE using Apache Ant with the build specification found at `<Install Dir>\SourceCode\Engine\DataRecordManager\QuickstartJavaSample\Java\JavaDrm\build.xml`. Or, it can be built using Eclipse and the supplied project files for Eclipse as follows:
 - a. Open Eclipse and choose *File\Switch Workspace\Other*. Enter `<Install Dir>\SourceCode\Engine\DataRecordManager\QuickstartJavaSample\Java` as the workspace path and click *Ok*. Eclipse will restart with your new workspace.
 - b. Open the *Import* tool found under the *File* menu. Select *General\Existing Projects into Workspace* and click *Next*. Enter `<Install Dir>\SourceCode\Engine\`

`DataRecordManager\QuickstartJavaSample\Java\JavaDrm` as the project root directory. The *Projects* area of the dialog should now contain *JavaDrm*. Click *Finish* to import the project into your workspace.

When you have imported the JavaDRM project into your Eclipse workspace, right-click on *build.xml* and select *Run as an Ant Build* to build the Java DRM layer. The resultant JAR file, called *JavaDRM.jar*, will be placed in `<InstallDir>\Binaries\Exe\{Configuration}`.

3. At run time, the **JniDrmReflmpl** library creates a JVM in which to run the Java DRM code. It will search the Java Classpath for *JavaDRM.jar*. To make sure that it can find your driver, copy *JavaDRM.jar* from the output directory to your classpath. The default is: `C:\Program Files\Java\jre6\lib\ext`.
4. The **JniDrmReflmpl** library will attempt to create a JVM using the *JVM.dll* library. Make sure that your *PATH* variable points to the location where the *JVM.dll* file can be found. The default location is: `C:\Program Files\Java\jre6\bin\client`.

Preparing and Using the Java Quickstart ODBC Driver

SimbaEngine SDK is installed with a sample database consisting of CSV files. This is intended for use by the Quickstart C++ sample driver and the Java-coded ODBC driver. Look for it in the `<Install Dir>\SourceCode\Engine\DataRecordManager\QuickstartSample\textdata` folder. It contains a number of files you can experiment with.

The easiest way to access this database for testing is to use the registry files included with SimbaEngine SDK to set up ODBC DSNs that point to the correct folder. Follow these steps to set up the DSNs.

1. Navigate to the `<Install Dir>\SourceCode` folder and open the *sample_engine.reg* registry file with a text editor. Modify the *sample_engine.reg* registry file to replace all the instances of "`<InstallDir>`" with the path of the installation directory. Be sure to use double back-



slashes \\ in path names where you want one back-slash to appear in the registry setting.

2. Run the modified *sample_engine.reg* registry file by double-clicking on it. This will create the DSNs you want.

You should now be able to run Excel and retrieve the test data in the CSV files. Run Excel and click on the *Data* tab (for Excel 2007). In the *Get External Data* group, choose *From Other Sources* and click on *From Microsoft Query*. Select the *Quickstart Java Sample Debug DSN* in the *Choose Data Source* dialog box and click *OK*. Pick a table from the *Query Wizard – Choose Columns* dialog box and click *Next*. Click *Next for Filter Data and Sort Order*, and click on *Finish* in the *Finish* dialog box. The *Import Data* dialog box will appear asking you where to put the data. Click *OK*. After a moment, your CSV data should populate your spreadsheet. Congratulations! You have retrieved database data from a Java data source directly into Excel using SimbaEngine SDK!

What's Next?

You can use the SimbaEngine SDK Java-built ODBC driver as an example when you write a new Java-based driver, or it can be modified and expanded until it does what you need. Java data is becoming more common, especially where web servers are running business applications for end users. The SimbaEngine SDK Java-built ODBC driver makes it easy to connect ODBC applications to your Java data source, opening up your data for purposeful and powerful analysis.

About Simba Technologies Inc.

Simba Technologies Inc. is the recognized world leader in standards-based data access products and solutions. Simba works with the world's leading software companies to deliver first class data connectivity solutions.

Simba is a pioneer in ODBC, MDX, ODBO and XMLA. Since 1991, Simba has developed advanced data access solutions for thousands of end users. Today, more than half of all MDX providers have been built with Simba technology, and through a partnership with Microsoft, Simba's SQL technology has been installed on more than 30 million desktops worldwide.

©2009 Simba Technologies Inc. All Rights Reserved.
Printed in Canada.

Simba Technologies Incorporated

938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel. +1.604.633.0008
Fax. +1.604.633.0004
Email. solutions (at) simba.com
www.simba.com

Simba and the Simba logo are trademarks of Simba Technologies Inc. All other trademarks or service marks are the property of their respective owners.