

## Re: Unable to load JdbcOdbc library

**Source:** <http://coding.derkeiler.com/Archive/Java/comp.lang.java.databases/2004-10/0089.html>

---

**From:** Daniel Dittmar ([daniel\\_at\\_dittmar.net](mailto:daniel_at_dittmar.net))

**Date:** 10/13/04

Date: Wed, 13 Oct 2004 00:13:55 +0200

SteveE wrote:

- > *Actually, there is a solaris ODBC driver, as I mentioned in my*
- > *previous post. Also, I'm very familiar with the native JDBC method of*
- > *doing things, the problem is as I mentioned that our database doesn't*
- > *expose a JDBC interface, only an ODBC interface... hence the problem.*
- > *Using ODBCTEST program that comes with the Solaris ODBC driver, I can*
- > *connect to the DSN without problems.*

You must distinguish between a ODBC driver (which you've got) and the ODBC driver manager (which you haven't).

On Windows, a program accesses the ODBC driver manager. Depending on the DSN, the driver manager loads the DLL for the matching driver and routes all calls to this driver.

Although driver managers exist for Unix (UnixODBC, iODBC), there are not standard, so there is few commercial software written to access them. Instead, you get a library and header files that conform to the ODBC spec. You can now write programs that use ODBC to connect to a specific database, in your case Informix. But without a driver manager, you cannot write a program that accesses arbitrary databases for which a driver is installed. And the JdbcOdbc bridge would be such a program. That's why SUN hasn't included the bridge in Unix versions of Java.

As about 'exposing a ODBC interface': technically speaking, databases expose network protocols. ODBC is then just a C library that makes the network protocol accessible.

As IBM advertises the JDBC driver as type 4 (<http://java.sun.com/products/jdbc/driverdesc.html>), there are several possibilities:

- the JDBC driver uses the same network protocol as the ODBC driver, in which case it is usable for your problem
- the JDBC driver uses a newer version of the network protocol, you're out of luck unless you can upgrade
- the JDBC driver uses a different protocol that was easier to implement in Java. You have to start a daemon on the server side, this daemon

comp.lang.java.databases: Re: Unable to load JdbcOdbc library

translates the request to the native format. In this case you are right that the database doesn't expose a JDBC interface.

Then there is the possibility that you can actually connect using 100% JDBC, but you want to use a 'native' driver because it has additional features like using your Solaris user as the database login. In that case you can ignore anything I said about network protocols.

Daniel