ASE ODBC Driver For UNIX And IQ Server

By Paul Vero

Paul Vero has worked at Sybase for 15 years. He is a Principle Product Support Engineer in Broomfield, CO. supporting various Sybase Connectivity and Middleware products. He can be reached at paul.vero@sybase.com.

The idea to port the ASE ODBC Driver to the UNIX Platform started years ago but never materialized until this year when a large demand developed for ASE connectivity to Sybase IQ server. These drivers are now accessible to any developer who dares to write some ODBC code on platforms such as Solaris, AIX or HP-UX. It started with the SDK 15.5 ESD#05, released in June 2010, with ASE ODBC driver build 15.5.0.1016.

The installer copies the driver into the *DataAccess64\ODBC* folder, including a Driver Manager (unixODBC). Anyone familiar with the LINUX ASE ODBC driver (see ISUG Technical Journal July-August 2009 "An ASE ODBC Driver on Linux") will find the configuration and usage of these drivers very similar; however, the use of the iAnywhere ODBC Driver Manager provides the means for Sybase IQ to connect to ASE with the ODBC driver. This article is a quick overview, providing some tips and pointers to get you started in the right direction.

The Platforms

The decision was made to support only 64-bit platforms for these ports. These are the new supported platforms:

- **Sun Solaris x86-64 64-bit**
- **Sun Solaris SPARC 64-bit**
- **HP-UX Itanium 64-bit**
- **IBM AIX 64-bit**
- **Linux on POWER 64-bit**

These platforms add on to the already existent Windows (x86 and x64) and Linux Intel based (x86 and x64).

Driver Manager

As mentioned, the drivers are bundled with the iAnywhere ODBC Driver Manager based off the unixODBC 2.14 Driver Manager. This is included on the UNIX platforms as the Linux systems typically include the Driver Manager. When installed, the files are contained in the *dm* directory, which contains the *include* and *lib64* folders. The contents from a Solaris Sparc install are displayed in **Listing 1** and **Listing 2**.

```
include:
  117697 Mar 23 13:11 odbc.h
   2106 Mar 23 13:00 poppk.h
   2234 Mar 23 13:00 pshpk2.h
   1900 Mar 23 13:11 saodbc.h
   4644 Mar 23 13:00 sqlos.h
   495 Mar 23 13:11 unixodbc.h
```

Listing 1: The Include Files

```
lib64:
  lrwxrwxrwx  1 sybase      17 Jun 15 21:48 libdbicu11.so -> ./libdbicu11.so.1
  -rwxrwxrwx  1 sybase 2504872 May 19 10:40 libdbicu11.so.1
  lrwxrwxrwx  1 sybase      19 Jun 15 21:48 libdbicu11_r.so -> ./libdbicu11_r.so.1
  -rwxrwxrwx  1 sybase 2515352 May 19 10:40 libdbicu11_r.so.1
  lrwxrwxrwx  1 sybase      19 Jun 15 21:48 libdbicudt11.so -> ./libdbicudt11.so.1
  -rwxrwxrwx  1 sybase 4353128 May 19 10:44 libdbicudt11.so.1
  lrwxrwxrwx  1 sybase      17 Jun 15 21:48 libdbodm11.so -> ./libdbodm11.so.1
  -rwxrwxrwx  1 sybase  878248 May 19 10:36 libdbodm11.so.1
  lrwxrwxrwx  1 sybase      19 Jun 15 21:48 libdbtasks11.so -> ./libdbtasks11.so.1
  -rwxrwxrwx  1 sybase   60904 May 19 10:50 libdbtasks11.so.1
  lrwxrwxrwx  1 sybase      21 Jun 15 21:48 libdbtasks11_r.so -> ./libdbtasks11_r.so.1
  -rwxrwxrwx  1 sybase   82192 May 19 10:50 libdbtasks11_r.so.1
  lrwxrwxrwx  1 sybase      15 Jun 15 21:48 libodbc.so -> ./libdbodm11.so
  drwxrwxrwx  2 sybase     512 Jun 15 21:48 res
```

Listing 2: The lib64 Directory

"These drivers are now accessible to any developer who dares to write some ODBC code on platforms such as Solaris, AIX or HP-UX."

Don't worry too much about the details yet as next we cover the installation and go over the environment variables required for library access and ODBC related matters.

Installation

The installation is performed by the SDK 15.5 installer, InstallAnywhere. When asked to choose the products, make sure the "ASE ODBC Driver" option is selected. The typical list of products to install looks like this:

1- [-] Open Client
2- |-[X] DB-Library
3- |-[X] Embedded SQL/C
4- |-[X] Embedded SQL/Cobol
5- |-[X] XA Interface Library for ASE Distributed Transaction Manager
6- |-[ ] Additional Connectivity Language Modules
7- [X] jConnect 6.0 for JDBC
8- [X] jConnect 7.0 for JDBC
9- [X] ASE ODBC Driver

I choose the directory */sybase/ocs/syb155* to install, setting my SYBASE environment variable to this location. The driver directories are located under *$SYBASE\DataAccess64\ODBC*, containing *dm* (Driver Manager), *lib*, *samples* and *sp*. The contents of the *lib* directory are shown in **Listing 3**.

```
lib:
  -rwxr-xr-x  1 pvero   1387768 Jun  7 17:15 libsbgse2.so
  -rwxr-xr-x  1 pvero   6978528 Jun  7 17:15 libsybcsi_certicom26.so
  -rwxr-xr-x  1 pvero   4139704 Jun  7 17:15 libsybcsi_certicom_fips26.so
  -rwxr-xr-x  1 pvero    636152 Jun  7 17:15 libsybcsi_core26.so
  -rwxr-xr-x  1 pvero   2527544 Jun  7 17:15 libsybcsi_openssl26.so
  -rwxr-xr-x  1 pvero     26520 Jun  7 17:15 libsybcsi_profiler26.so
  -rwxr-xr-x  1 pvero   2116968 Jun  7 17:15 libsybdrvodb.so
  drwxrwxr-x  5 pvero       512 Jul  8 09:48 locales
```

Listing 3: The lib Directory

The ODBC driver, *libsybdrvodb.so*, lives here with the *csi libs* required files for password encryption. The *samples* contain some C++ source code to build and test for connectivity and the *sp* directory contains the stored procedures required for metadata access for the ODBC CSPs.

Under the $SYBASE root directory, scripts are created to provide the environment but there are some important items missing and I'll cover this in the next section. At this point, the driver is installed and ready to be integrated into your IQ server environment. Once in place, IQ server can be used to create proxy tables to your ASE database.

Configuration

Before integrating to IQ, a few things need to be covered. These are the environment variables if you're interested in building applications. In my example, I’m using the Solaris Sparc platform. Here are the variables I create:

```
sentenv LD_LIBRARY_PATH $SYBASE/DataAccess64/
ODBC/dm/lib64:$SYBASE/DataAccess64/ODBC/
lib:$LD_LIBRARY_PATH

setenv ODBCINI /sybase/ocs/syb155/odbc.ini

setenv DM_LIBRARY_PATH /sybase/ocs/syb155/DataAccess64/ODBC/dm/lib64

setenv DRIVER_LIBRARY_PATH /sybase/ocs/syb155/
DataAccess64/ODBC/lib

setenv INCLUDE_PATH /sybase/ocs/install/unixODBC/
usr/local/include
```

I installed my own unixODBC so my INCLUDE_PATH uses the headers included in the package. In your case, you can use the files in the *dm/include* directory and follow the instructions in the cover letter to softlink the *unixcodbc.h* file to the appropriate ODBC headers: *sql.h*, *sqlext.h* and *sqltypes.h*.

You need to manually create the *odbc.ini* file, and I created mine in the $SYBASE folder. This file must reside here for Sybase IQ to access it. The contents of the file is shown below:

```
[pvlnx1503]
Description = Sybase ODBC Data Source
UserID      = sa
Driver      = /sybase/ocs/syb155/DataAccess64/
ODBC/lib/libsybdrvodb.so
Server      = pvero-rhel5
Port        = 1503
Database    = pubs2
UseCursor   = 0
```

IQ Integration

As you notice, I keep talking about integrating the driver to the IQ environment. The ASE ODBC Driver for UNIX and Linux platforms is only released in the SDK 15.5, not bundled with Sybase IQ. Therefore, before you can create proxy tables in IQ, you need to add certain environmental variables to your IQ environment.

I used IQ 15.2 on Sun Solaris, 64-bit. Since I use C shell most of the time, I modified the *IQ-15_2.csh* script found in the $SYBASE/IQ-15_2 folder. Assuming my Sybase root location is */sybase/iq/syb152*, this is what I added to the script:

```
setenv ODBCINI "/sybase/iq/syb152/odbc.ini"

setenv LD_LIBRARY_PATH "/sybase/ocs/syb155/DataAccess64/ODBC/dm/lib64:/sybase/ocs/syb155/
DataAccess64/ODBC/lib:${LD_LIBRARY_PATH}"
```

This provides IQ with the location of the iAnywhere Driver Manager, ASE ODBC driver and odbc.ini. Now I need to add information to IQ server so I can connect to the ASE. My DSN name is “pvsun155”. To make this work, you need to create a server on IQ like this:

```
CREATE SERVER asetest
CLASS ‘aseodbc’
USING ‘DSN=pvsun155;UID=sa;PWD=;DATABASE=od
bc;’
```

This creates the server, *asetest*, using the ASE ODBC Driver, as designated by the class of ‘aseodbc’. The next step is to create a proxy table. This is done with the following syntax:

```
create existing table iq_ado_table at ‘asetest.odbc.dbo.
ado_table’
```

Now that I have a proxy table that connects to the ASE database, I can retrieve data to the IQ server. I can also insert records. The select statement is as follows:

```
1> select * from iq_ado_table
2> go
 c1        c2                                   c3
      c4
 ----------- ----------------------------------------------
-- -------------
       ----------------------------
        1 Winter                               123.0000
       Mar  1 2010 12:00:00.000000AM
        2 Spring                               456.0000
       Jun  1 2010 12:00:00.000000AM
        3 Summer                               789.0000
       Sep  1 2010 12:00:00.000000AM
        4 Fall                              101112.0000
       Oct 10 2010  9:00:00.000000PM

(4 rows affected)
```

To insert records, I can execute an insert statement as follows:

```
insert into iq_ado_table values (5, 'IQ insert', 2312,
getdate())

1> select * from iq_ado_table
c1      c2                              c3
     c4
 ---------- ---------------------------------------------
-- -------------
1
      5 IQ insert                             2312.0000
```

Very cool, right? If I want to send a native statement, I can use the FORWARD TO statement to execute a query directly:

```
FORWARD TO asetest "select * from odbc..ado_table"
```

Keep in mind if I didn't specify a database in my ODBC connection string when I defined the server definition, I need to qualify this when I issue queries if the target object resides in a non-default database.

To show you how important it is to have everything just right, if you don't have the ODBCINI environment variable available to the IQ server, accessing the *asetest* server will throw the following error:

```
szSqlState = "HY000", *pfNativeError = -656,
MessageText = "[Sybase][ODBC Driver]
[Sybase IQ]Unable to connect to server
'asetest': [Sybase][ODBC Driver Manager]
Data source not found and no default driver
specified"
```

It's the second message that tells me the ODBCINI isn't set properly. At the manager level, it's really telling the truth when it can't find the data source – it can't locate the *odbc.ini* file!

If my ASE ODBC Driver library path, like */sybase/ocs/syb155/DataAccess64/ODBC/lib*, is not set in LD_LIBRARY_PATH, I get this error [from an isql connection to IQ]:

```
Msg 11206, Level 16, State 0:SQL Anywhere
Error -656: Unable to connect to server
'asetest': [Sybase][ODBC Driver][SQL Any-
where]Specified database not found
```

Sort of cryptic, but keep in mind the IQ server is connecting to the ASE with ODBC API calls, sort of like an application. IQ runs from an ASA engine. This engine is using ODBC calls to make things happen with the backend. In this case, when trying to locate the ODBC driver (ASE Driver) for the connection to *asetest*, nothing doing, the "Specified database not found" error is thrown, meaning the driver library isn't found.

Lastly, suppose for some weird reason the Driver Manager isn't found. In most cases, the IQ server has this available in its own *lib64* directory, but suppose the *libodbc.so* library isn't available. When I run the remote query as follows:

```
1> select * from iq_ado_table
2> go
```

The IQ server will throw this error:

```
Msg 11206, Level 16, State 0:SQL Anywhere
Error -656: Unable to connect to server
'asetest': [Sybase][ODBC Driver Manager]
Unable to load driver/sybase/ocs/syb155/
DataAccess64/ODBC/lib/libsybdrvodb.so
```

Conclusion

Now you should be able to install and configure the ASE ODBC Driver on your UNIX platform and use it with IQ Server. If you create the environment variables in a script, either the SYBASE IQ script or your own, then running the IQ server with ASE ODBC Driver will be a cinch. In a future article, I'll show you how you can build and execute your own applications. This is where you can use the ASE ODBC Driver in a standalone situation, going beyond using it with IQ. For now, enjoy running the ASE ODBC Driver on your new platform. ■

References

unixODBC – if you dare to install your own ODBC Driver manager *http://www.unixodbc.org/*

Link to SDK 15.5 New features guide – look under ESD #5 for driver info: *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.help.sdk.15.5/title.htm*