Platform Confusion with the JDBC-ODBC-Bridge

I had a plain old JAVA application using Sun‘s JDBC-ODBC-Bridge, which failed on Windows 7 (64 Bit) with the error message „The specified DSN contains an architecture mismatch between the Driver and Application“. It took me quite a while to sort out all of the 32/64 confusion that arises.

64-bit Windows Does not Imply 64-bit MS Office

The first problem I encountered, was creating an ODBC Source for the ACCDB file that I wished to access. Under *Control Panel / Administrative Tools / Data Sources (ODBC)* I could not find a driver for MS Access, although I hat MS Office 2010 installed. After a while I found out, that MS Office 2010 comes in two flavors: a 32-bit version and a 64-bit version. If you execute the *setup.exe* in the root folder of the MS Office 2010 distribution DVD, then the 32-bit version gets installed – even though you are running the setup on a 64-bit architecture with 64-bit Windows installed.

32-bit ODBC Data Sources in 64-bit Windows

There is actually a way, to install a 32-bit ODBC Data Source in Windows 7. The *Data Sources (ODBC)* administrative tool executes the 64-bit ODBC data source administrator *C:\Windows\system32\odbcad32.exe*. (Never mind the „32“ in the name: on a 64-bit platform, this is a 64-bit executable!) To get the 32-bit ODBC Data Source administrator, one needs to run the *C:\Windows\SysWOW64\odbcad32.exe* (Never mind the „64“ in the name of the folder: this is a 32-bit executable, for WOW64 means „Windows (32) On Windows 64“. The name reminds us of the WOW layer in Windows 95, which eased the transition from 16-bit to 32-bit Windows. It consisted of 16-bit stubs for legacy 16-bit programs that needed to run on 32-bit Windows.)

If you have the 32-bit MS Office installed, you will be able to install the usual *MS Access Driver (*.mdb)* as a 32-bit ODBC data source. This driver is the good old 32-bit jet engine executable *ODBCJT32.dll*.

Installing 64-bit MS Office 2010

The MS Office 2010 DVD actually contains the two folders *x86* (32-bit – never mind the „86“ which refers to the Intel architecture!) and *x64* (64-bit). The two versions of MS Office cannot coexist. It is necessary to uninstall MS Office 2010 (32-bit) first, before MS Office 2010 (64-bit) can be installed. Just run *x64/setup.exe* to install the 64-bit version.

If you have the 64-bit MS Office installed, you can install either truely 64-bit ODBC data sources using the initial *Control Panel / Administrative Tools / Data Sources (ODBC)* approach which executes *C:\Windows\system32\odbcad32.exe* or 32-bit ODBC data sources running *C:\Windows\SysWOW64\odbcad32.exe*. In this case you will notice in both administrative panels the additional *MS Access Driver (*.mdb, *.accdb)*. This refers to the truly 64-bit *ACEODBC.dll* which replaces the Jet Engine. Both types of datasources show up in both administrative panels (probably to prevent name clashes) but the 64-bit ones cannot be administered in the 32-

bit ODBC administration panel. The name of the new 64-bit ODBC driver (*MS Access Driver (*.mdb, *.accdb)*) is a bit confusing. Already the 32-bit version could handle ACCDB files. The file ACCDB type was introduced with MS Office 2007 as the OOXML replacement of the old MDB format, is independent of the platform and can be read and written by 32-bit MS Access as well as 64-bit MS Access.

64-bit Windows Does Not Imply 64-bit JAVA

Under 64-bit Windows it is quite possible to install either 32-bit JAVA or 64-bit JAVA or both. Most binary JAVA classes will run under either. It seems to make more sense, to install 64-bit JAVA. There is only one area, where the platform-dependency creeps in: When a JNI (JAVA Native Interface) link to a native code shared object (*so*) or dynamic link library (*DLL*) is established, it matters, whether the library is a 32-bit or a 64-bit executable.

And this is precisely the case in the JDBC-ODBC-Bridge. The *jre/bin/JdbcOdbc.dll* in the 32-bit version of JAVA is a 32-bit DLL. The *jre/bin/JdbcOdbc.dll* in the 64-bit version of JAVA is a 64-bit DLL.

At this point I needed a fast way to determine the platform of a DLL or EXE and created a small (32-bit) utility *platform.exe* which will analyze the PE file format of a DLL or EXE and display the platform it was compiled for. It is available for download here together with its source code.

The class *sun.jdbc.odbc.JdbcOdbc.class* loads the native library *JdbcOdbc* from the *java.library.path* which points to the *jre/bin* folder of the JAVA instance with which the JAVA application was started.

64-bit JNI Can Only Interface to 64-bit Native Libraries...

... and 32-bit JNI only to 32-bit Native Libraries!

So if you start a JAVA program connecting to a 32-bit ODBC data source using a 64-bit JAVA instance it will fail. Similarly starting a JAVA program connecting to a 64-bit ODBC data source will not succeed.

You will just have to remember, to synchronize the JAVA platform and the ODBC Data Source administrator. Either use 32-bit versions for both or use 64-bit versions for both.

Why it is Useful to Have a 32-bit JAVA Version Installed

As mentioned before, the JAVA byte code is the same for 32-bit or 64-bit JAVA. Strictly speaking there is only one kind of JAVA. The only place where the platform matters is in the interface to native code. This is used quite extensively by some JAVA code. For example LibreOffice/OpenOffice require 32-bit JAVA.

It is impossible to support both kinds of ODBC connections simultaneously, because the ODBC API class has the same name in both 32-bit JAVA and 64-bit JAVA and can only be loaded once – unless some very problematic ClassLoader tricks are used.

How to Detect the Installed JAVA Versions

Usually one finds the “current“ JAVA version in the registry under *HKLM\Software\JavaSoft\Java Runtime Environment*, where the value *CurrentVersion* indicates, which subkey to examine for the value *JavaHome*. But beware, if your executable accessing the registry is an old executable targeted at 32-bit Windows: Then all path names starting with *C:\Program Files* are redirected to *C:\Program Files (x86)*! Using a 64-bit program to access the registry, this redirected 32-bit version can be found under *HKLM\Software\Wow6432Node\JavaSoft\Java Runtime Environment*.

If one wants to detect the installed JAVA version automatically one will most likely make use of reg.exe. On the 64-bit platform this has changed its output format a bit. It replaces tab characters by four blanks and umlauts by multi-byte sequences. The same effect could be achieved on the 32-bit platform by running its reg.exe with the unicode switch /u:

```
cmd /u /c reg query „HKLM\Software\JavaSoft\Java Runtime Environment“ /v CurrentVersion
```

Hartwig Thomas 17. November 2011