Access on 64bit ColdFusion? Yes we Can!!

Posted At : December 31, 2010 1:51 PM | Posted By : Mark Kruger
Related Categories: Coldfusion & Databases

First, let me make sure I do not get off on the wrong foot here. I am most explicitly *not* saying that you should be using Access on a production server with heavy traffic as a regular application data source. In my opinion Access should be limited to the smallest use possible on web servers. It's fine if you have a small intranet application to keep track of equipment or handle tickets among 4 or 5 people - although why you wouldn't at least move to MySQL or MSSQL express is puzzling. However (and the Muse can't be clearer than this) if you are using Access to backstop a high traffic ecommerce site or content management for an important customer then shame on you. It's a problem you will have to solve - maybe not today, maybe not tomorrow... but someday and for the rest of your product life cycle.

Having said all that you might remember my post on setting up an **Access Proxy**. The problem I was solving in that case was quite specific. I was using Access as an export format for an enterprise application. Users would run a very large report and then have the choice to export it to Excel or Access for offline viewing or mining. As far as I'm concerned this is an appropriate use of Access - as an offline data mining tool. My process basically copied the file to a unique location and used the **Access Passthrough** method to populate it, zipped it up and sent it to the users browser.

When I moved to 64bit ColdFusion I was forced to come up with an elaborate proxy service to insure that the process continued to function as usual. This was because I could not figure out how to get Sequelink (those funky "ODBC Server" and "ODBC Agent" services that install with ColdFusion) to talk to ColdFusion any more in the 64bit environment. Recently however, Brent Fry delivered this **excellent post** as well as provided me with some additional info for the windows 2003 server platform. So I now have a new process that is capable of synching up Sequelink with ColdFusion.

There's nothing wrong with Brent's post of course, but I have a desire to codify this in my own words. Plus I know I'll need it again and if I don't write it down for myself I forget where I found it the last time (no offense to Brent and cfexecute.com and with all credit due!). With that in mind, here's the Muse version of Access on 64bit ColdFusion with a little "under the hood" stuff thrown in.

Finding the Jet Driver

To start we need to locate those pesky 32 bit drivers. It's important to know that the issue with the venerable JET driver (the long-in-the-tooth driver we have using for decades to access MS Access files) on 64 bit Windows is not that it doesn't work on the 64bit OS. It's simply that there is no *64 bit version* of Jet. When you go to "Administrative Tools" and click on "Data Sources (ODBC)" on a 64bit Windows box you are seeing *only the 64bit drivers*. It's actually really goofy. The link in administrative tools points to %SystemRoot%\system32\odbcad32.exe. The name of that file would *seem* to indicate 32 bit drivers to me, but in reality what you are seeing is the 32 bit control panel applet for managing system DSN's. The list of drivers available (if you click on Add) will be only 64bit drivers. By default this means you will likely see only MS SQL server (installed by default) and any other 64bit ODBC driver you have installed.

Here's the catch. Windows 64bit *is capable of running 32bit software*. You know this if you think about it. In fact the odbcad32.exe file above is a *32 bit* control panel applet for managing 64 bit drivers. So the trick is to figure out how to manage 32bit drivers on your system. The good news is that the *old* version of the ODBC manager (the one that handles 32 bit drivers) still exists on your 64bit machine. Take a look in the %systemroot%/sysWOW64/ folder. You will see a version of the same file - odbcad32.exe. Open this control panel applet and you will discover that it actually manages the old list of *32 bit drivers*. I know it's convoluted - a file in "system32" called odbcad32.exe manages 64 bit drivers while a file in the sysWOW64 folder manages 32 bit drivers. If you need to know, the "sysWOW64" folder stands for "Windows on Windows 64" - meaning plain old Windows 32 bit aps running on the Windows 64 bit platform. Now some Muse readers just lit up with delight at the new information while other's had their eyes glaze over with a dull light from TMI. Snap out of it and let's keep that tool up while we talk about the next thing - the Sequelink server.

Sequelink Server Fun

In all MX version of CF (starting way back with version 6 when Ben Forta was still a lad and Sean Corefield was still tending sheep) MS Access has been handled using the "Coldfusion MX ODBC Server" and the "Coldfusion MX ODBC Agent". These 2 services are running a version of a product called "Sequelink Server" from Merant. Under the hood the Sequelink server sets up ODBC links to the Access file and provide a socket interface for JDBC. CF makes JDBC calls to the socket and Sequelink translates the calls to ODBC, gets the result, retranslates to JDBC and sends the result back upstream to the JVM.

Incidentally this is one reason why ODBC and Access is even *less* of a good idea in ColdFusion MX than it was in ColdFusion 5. At least in ColdFusion 5 (running as native C++ code on windows) CF accessed the ODBC subsystem directly and used ODBC drivers. As soon as CF became a Java product an intermediate layer - a "bridge" layer - was introduced to make communication between JDBC and ODBC possible. This intermediate layer further complicates the use of ODBC and adds additional latency to the system already under pressure (Access never worked that great on CF 5 or any production platform for that matter - it's simply not a database "system". It is, rather, a file and driver pair). In any case this "bridge" layer is provided by the Sequelink server. If you look in the task manager it shows up as the swsoc.exe process.

Now putting two and two together we can assume that, since Sequelink uses a socket interface for CF to access and since the 32bit drivers *do indeed* exist on Windows 64bit - it stands to reason that it must be possible to get the two of them to work together. After all - Verity is 32bit and it cooperates with CF just fine on the same principle - a socket type interface. So this *must be* possible - right?

The Real Problem

The real problem here is actually pretty basic. It's not that the driver doesn't work or doesn't exist. The same 32bit drivers that have always worked will work with the same Sequelink server that has always worked. Instead the real problem is that the registry entries, driver paths and other "configuration" stuff under the hood is now in different places. The trick is to mirror the *old configuration* somehow.

The Fix - Windows 2008r2

Step 1:

The fix is different depending on whether you are running 2008r2 (which *only* comes as 64 bit) or Windows 2003 64bit. Let's start with r2 because the fix is easier for that platform. Remember the odbcad32.exe file in the /sysWOW64/ folder? Start with that little applet and click on the "system DSN" tab. Click on the "Add" button and you will see a list of drivers including the "MS Access" driver. Enter the same name for the data source that you intend to use in the ColdFusion administrator as the data source name. Browse to the Access file you are going to use and click ok. You should now see your ODBC data source in the list.

Step 2:

Next, log into the CFIDE Administrator and add a new "MS Access" data source using the same name as you did in the ODBC control panel applet. When you click "add" you will get an error message that says "Unable to Update the NT registry", but the data source will still register. To test, you can try running a simple query like this one:

```
<cfquery name="check" datasource="newAccessDsnon64">
    SELECT now() AS dt
</cfquery>
```

If this works then you are ready to go. But before we leave Windows 2008r2 let's take a look at what happened when you added that entry to the odbcad32.exe applet in sysWOW64. When you did this you actually added an entry to the registry at HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC. This is what knits together the Sequelink server with the file and allows a successful connection.

The Fix - Windows 2003 Server 64bit

If you are using Windows 2003 things are a little trickier. Again it was Brent Frye pointing me in the right direction. What's happening in 2003 is that the registry entries for the 32 bit drivers are not precisely mirrored. If you try the procedure above on a 2003 64bit server you will find that the HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC node is NOT created - so Sequelink cannot find the file and driver. Instead the *old style* registry entry is created at HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\. The entries are *correct* - they are just in the wrong place. What's needed is a little creative registry work. Please note - if you are reading this blog and have gotten this far you are likely an experienced developer or sys-admin. All the usual warnings about editing the registry apply here. Yes Virginia you can make your server completely useless if you don't know what you are doing. That said - here are the relatively painless instructions.

Step 1.a

After step one but *before you log in to the ColdFusion Admin*, go to the registry editor and find the entry at the HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\*data source name* entry. For example, I added an system DSN called bob so I navigated to HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\*bob*. Export this key to a .reg file and open it for editing in notepad. You should see something like this (with paths and dsn name reflecting your own entry of course).

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bob]
@=""
"Description"=""
"DBQ"="D:\\dbs\\uncles\\bob.mdb"
"SystemDB"=""
"UID"=""
```

```
"PWD"=""
"DriverId"=dword:00000019
"FIL"="MS Access"
"DefaultDir"="D:\\dbs\\uncles\\"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bob\Engines]
@=""

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bob\Engines\Jet]
@=""
"PageTimeout"=dword:00000258
"MaxBufferSize"=dword:0000fa00

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\bob\Engines\Jet 2.x]
@=""
"PageTimeout"=dword:00000258
"MaxBufferSize"=dword:0000fa00
```

Carefully change the first part of the path of each of the [HKEY_LOCAL_MACHINE...] entries to reflect *this* path instead:

```
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC\
```

Your finished script should look similar to this:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC\ODBC.INI\bob]
@=""
"Description"=""
"DBQ"="D:\\dbs\\uncles\\bob.mdb"
"SystemDB"=""
"UID"=""
"PWD"=""
"DriverId"=dword:00000019
"FIL"="MS Access"
"DefaultDir"="D:\\dbs\\uncles\\"

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC\ODBC.INI\bob\Engines]
@=""

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC\ODBC.INI\bob\Engines\Jet]
@=""
"PageTimeout"=dword:00000258
"MaxBufferSize"=dword:0000fa00

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432NODE\ODBC\ODBC.INI\bob\Engines\Jet 2.x]
@=""
"PageTimeout"=dword:00000258
"MaxBufferSize"=dword:0000fa00
```

Save the file with the .reg extension and run it to insert these entries. Now Sequelink can "find" the files and proper drivers and you can add the data source using the ColdFusion admin (Step 2).

Conclusion

This is a fairly involved process and I would not want to implement it on a server with dozens of Access files. As always the Muse believes in minimal and judicious use of Access on production web servers. Still, if you absolutely need to get Access/Jet running in a 64bit Windows environment, it's nice to know it's still possible. Now, beloved Muse readers, a word of caution - please keep comments relevant to the topic. I'm not afraid of criticism or correction (as my readers well know) but I rather abhor the kind of conflagrations that sometimes occur when discussing MS products (or windows or mac or linux or adobe etc etc). Keep your comments civil and on point please. Let's make entries like this helpful. I often refer folks to my blog with the cavaet that "some of the best tips are in the comments", so help me make sure that stays true.