

# Re: Complicated Connection Problems bewteen ADP and SQL Server

---

*Source:*

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.adp.sqlserver/2006-02/msg00191.html>

---

- *From:* "Norman Yuan" <NotReal@xxxxxxxxxxx>
  - *Date:* Thu, 16 Feb 2006 12:48:10 -0700
- 

There is no magic here, only two things:

1. If the workstation can see (connect to the SQL Server);
2. If the user has appropriate permission to the database on that SQL Server.

My guess is that your problem might be the latter rather than former.

If a new workstation computer is set up normally and connect to the network, I do not see a special reason that the computer somehow cannot connec to the SQL Server. I set up so many computers in oour office, never see something special I have to do to run a program that access SQL Server on the network. However, SQL Server security is a bit complicated and it is never a simple thing like knowing the username/password pair.

You did not mention the SQL Server using Windows security or SQL Security. In your code, it seems that you use SQL Server security. A company of that sizs should have a normal domain network. You are supposed to use Windows security unless you hace speifc need for SQL Server security. There is also odd that you need a DSN for ADP. ADP is designed to connect to SQL Server directly, why you need a ODBC DSN?

"Geoffrey Barnes" <nospam@xxxxxxxx> wrote in message  
[news:Pu1Jf.19402\\$vU2.4293@xx](news:Pu1Jf.19402$vU2.4293@xx)

About 3 years ago, I was told to create a SQL Server 2000 database for a client of ours, and to set up an Access XP/2002 project (adp) front end that would be used on their network to interact with the data. This database needs to be updated every six months. Since there has been some "feature creep" in the front end, I have almost always needed to install a new adp front end on the 5-8 workstations where the users happen to sit.

I'm not a network guy by any means. This client has thousands of workstations in at least two different building, so they provided all the expertise for getting the user workstations talking to the SQL Server. Most

## Re: Complicated Connection Problems between ADP and SQL Server

of the time, I just walk in with a back-up of the new database, restore it onto their server, slap a new front end on the 5-8 workstations that will need to work with the database, and off I go.

The very first time that we ever did this installation, there were some problems getting the workstations to talk to the server. The guy I was working with did some magic on the workstations and the connections worked just fine after that. Then he moved onto a new job, and they assigned someone else to work with me during my twice-yearly visits. All of the workstations that had been used before would work just fine, the connections would go through without any effort on our part, and the installation was easy. But whenever one of the workers had been given a new computer, the connection would fail and the adp wouldn't be able to talk to the server. This wasn't a problem, though, since the guy I was working with also seemed to know what to do. He would get on the workstation, do his magic, the connection would be established, and that workstation would never again cause us any problems on any future visits.

This week I made my latest visit to the client's offices. The guy I had been working with us died unexpectedly a few months ago, and nobody there seems to have any idea what he did to make these virgin machines -- the ones that had never been used to work with this database in the past -- talk to the SQL Server. And of course, two of the five users that need to work with the database have recently received new computers.

Since I don't have anyone there who can fix the problem on their end, and since it was probably sloppy of me to require them to alter their workstation setup in the first place, I was wondering whether the problem might be in my ADO connection string. Maybe I could use a different string that would connect just fine, without any requirement to change any settings on the workstation.

The SQL Server at the client uses mixed-mode security. My project opens up with a custom login form to get the username and password. Then it calls the following function to establish the connection for the project. Note that I first open up a generic ADO connection (cnnTest) to the database to make sure that it works and to trap any errors that might come up. Then, if that test connection works, I go ahead and call CurrentProject.OpenConnection using the same string.

----- BEGIN VBA CODE -----

```
Public Function EstablishConnection(ByVal user As String, _
```

Re: Complicated Connection Problems bewteen ADP and SQL Server

```
ByVal password As String, _
Optional ByVal displayWarnings As
Boolean = True, _
Optional ByVal finalTry As Boolean =
False) As Boolean

Dim strServer As String
Dim strDatabase As String
Dim strConnect As String
Dim cnnTest As ADODB.Connection

Call DoCmd.Hourglass(True)

strServer = SERVER_NAME
strDatabase = "PbcPrimary"

strConnect = "Provider=SQLOLEDB;Data Source=" & strServer & ";Initial
Catalog=" & strDatabase & _
";Persist Security Info=FALSE"

Set cnnTest = New ADODB.Connection

On Error GoTo LoginFailure
Call cnnTest.Open(strConnect, user, password)
On Error GoTo 0

Set cnnTest = Nothing

'If things have progressed to this point, then cnnTest has been
successfully
established. Switch the
'CurrentProject's connection to this new connection.

Call CurrentProject.OpenConnection(strConnect, user, password)

EstablishConnection = True
Call DoCmd.Hourglass(False)

Exit Function

-----
-----

LoginFailure:

Call DoCmd.Hourglass(False)

If displayWarnings Then

If Err.Number = -2147217843 Then
```

## Re: Complicated Connection Problems bewteen ADP and SQL Server

```
If Not finalTry Then
Call MsgBox("The user name and password that you entered are not
valid. Please " & _
"check your entries and try again.", vbExclamation, "Login
Failed")
Else
Call Fatal("The user names and passwords that you have entered
were
not valid. If " & _
"you have forgotten your user name or password, please see the
local JRC site manager.", _
"Shutting Down")
End If

ElseIf Err.Number = -2147467259 And LTest(Err.Description, "Cannot open
database") Then

'Note that you get this error for at least two conditions. First,
this error occurs if the server is found,
'but the database cannot be. Secondly, it occurs if the user's login
exists on the server, but the login
'has not been given permission to access the database. Since the
first situation seems more likely, that is
'the problem that is being described in the MsgBox.

Call Fatal("The reconciliation database could not be found on the
database server.", _
"Database Missing")

ElseIf Err.Number = -2147467259 Then

Call Fatal("The " & strServer & " database server could not be
located
on the " & _
"network.", "Server Not Found")

Else

Call Unexpected("frmLogin, cmdOK_Click", Err.Number, Err.Description,
"trying to " & _
"connect to the reconciliation database", "Connection Failed")

End If

End If

EstablishConnection = False

End Function
```

----- END VBA CODE -----

As I said, this code works just fine on any of the "experienced" machines, but the new ones don't like it. On the "virgin" workstations, I get an error -2147467259, [DBNMPNTW]ConnectionOpen (CreateFile()). Because of my error-handling routine, this gets reported to the user as "The database server could not be located on the network".

I did some research on this error. It has something to do with "named pipes", and I'm sure many of you actually understand what it means. But I'm not a network guy. Moreover, this isn't my server. I can't just go into the server and change the settings to allow the use of named pipes. So I thought that I should instead change my connection string to something that the server and the network would accept.

I created a brand new adp on one of the virgin workstations, binded it to the target database on the SQL server, and found that I was indeed able to connect to the server and the database without any problem at all from within this new project. When I checked the connection string of this bound project, I found the following:

```
Provider=Microsoft.Access.OLEDB.10.0;Persist Security Info=False;Data Source=ClientServerName;User ID=MyUserName;Initial Catalog=PbcPrimary;Data Provider=SQLOLEDB.1
```

This string has me puzzled a bit. I don't understand why what I thought was the "Provider" has now become the "Data Provider", and I really don't understand why the provider is now something that is native to Access itself instead of the SQL Server OLEDB provider. But what was even stranger was what I found when I checked the BaseConnectionString property of the bound project:

```
PROVIDER=SQLOLEDB.1;PERSIST SECURITY INFO=FALSE;INITIAL CATALOG=PbcPrimary;DATA SOURCE=ClientServerName;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=MyWorkstationName
```

Even after reading the on-line help file, I still don't really understand what the difference is between CurrentProject.BaseConnectionString and CurrentProject.Connection.ConnectionString. Maybe someone here can enlighten me.

But here is where things really get weird. When I copied the working connection string from the bound project into my own project, it still didn't work. When I checked all of the connection properties (under the

## Re: Complicated Connection Problems bewteen ADP and SQL Server

File menu) of the working bound project, I found that it was using a different network library. I can't remember which one specifically, but it definitely wasn't "DBNMPNTW" and I think it was probably "DBMSSOCN".

And now things get really strange. I added in the command "Network Library=DBMSSOCN" into my connection string. When I did that, my initial test ADO connection (connTest) worked just fine. Success! But when the program then tried to use the exact same connection string in the CurrentProject.OpenConnection method, I got an entirely new error. Since I'm no longer at the client site, I don't have the exact text of the error, but it was something along the lines of "-2147467259 (8004005) Client unable to establish connection".

Why would a regular ADO connection work, but the project connection fail? Does anyone know of some connection string settings that might work here?

Alternately, maybe somebody knows or can guess what kind of workstation magic was done to allow the connection on the "experienced" computers. We did find one thing. All of the older machines have a DSN which points directly at the SQL Server. But the newer machines have no such DSN. So I

think it has something to do with the presence of this DSN on the older machines. But we couldn't create a DSN on the newer machines. We got the same "Client unable to establish connection" error when we tried. So while the DSN might be necessary, it's also the case that something else has to be done before the DSN can be created. Then again, the connection string I'm using is DSN-less, so maybe the DSNs were just there because the guy who fixed the workstations for me used DSNs to make sure his fix was working.