

Can't connect with ConnectionString (D7+ADO)

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2005-05/msg00139.html>

- *From:* Rick Osborne <NRickoSPAM@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 May 2005 04:54:51 -0500
-

Greetings.

I am driving myself crazy trying to get Delphi to work with ADO. I have a DLL that is loaded by a service (no GUI), and that DLL needs to do some database work. Ultimately, it'll do thread pooling and lots of other nifty stuff, but for now I can't even get it to talk to the database.

Goal: Allow the user to authenticate against a database given a connectionstring of their own making. This could be against anything from an Excel spreadsheet to MS SQL to Access, as long as ODBC/ADO/Whatever can see it.

Problem #1: All of the examples and tutorials assume that you can just drop a TADOConnection component on your form and go from there. Since this is a DLL, there is no form. I seem to have worked around this by doing the grunt work manually, but I am unclear as to if I am missing something. (See below.) Anyone know of any good tutorials for Delphi (or something similar) for mucking about with ADO without a GUI?

Problem #2: All seems to go well and good until I hit the db.Open(), at which point I get a "datasource name not found and no default driver" error. This is crap. I've tried countless variations on connection strings, including UDL files, and they all give the same error. Also, I notice that when I step through with the debugger after I set db.ConnectionString, but before the Open call executes, the debugger thinks ConnectionString is still an empty string. (Not inaccessible or optimized away, but an actual empty string.) Is there something else I need to do before I set the ConnectionString to make it stick?

I'm also not really married to ADO if someone can suggest an alternative that will cause me less headaches. As long as it meets the big requirements (no GUI, programmatically-settable connection properties, eventual thread pooling) I'm willing to try it.

```
// -----[ BEGIN PSEUDOCODE ]-----  
function Authenticate(var Username: String; var Password: String):  
    Boolean;
```

Can't connect withConnectionString (D7+ADO)

```
var
  db: TADOConnection;
  rs: _RecordSet;
  cmd: TADOCommand;
begin
  CoInitialize(nil); // <== Do I need this?
  cmd := nil;
  db := TADOConnection.Create(nil);
  try
    db.LoginPrompt := False;
    db.ConnectionString := _ConnString; // <== This is from a member
    db.Connection.Open(_Username, _Password); // <== Same here
    cmd := TADOCommand.Create(nil);
    try
      cmd.Connection := db;
      cmd.CommandText := _SQL; // <== Ever more member vars
      rs := cmd.Execute();
      try
        if not rs.EOF then begin
          rs.MoveFirst();
          // ... bla
        end
      end
    end
  end
```