

The Delphi Clinic

Edited by Bob Swart

Bring your problems to our panel of experts!

If there's something puzzling you about an aspect of Delphi, just email the Delphi Clinic Editor, Bob Swart, at Compuserve 100434,2072 or write or fax us at The Delphi Magazine

QI've just installed Delphi and I just can't get it to run right. I keep getting errors like:

```
Application Error. Exception
EDatabase Error in module
FormExpt.DLL at 0007:0EAC.
Failed to initialize IDAPI.
Error code $2C09.
```

What's gone wrong?

A Please see if you've got SHARE loaded in your AUTOEXEC.BAT. The correct line there should read (at least):

```
SHARE /F:4096 /L:40
```

Alternately, you could use:

```
device=vshare.386
```

in the [386Enh] section of the SYSTEM.INI file. Note that Delphi does ask Windows whether or not share is loaded, but Windows (incorrectly) always reports SHARE to be loaded, even if it isn't.

QI'm building a Delphi component and want to be able to tell if it's being used in design mode or at run time. How can I do this?

A Although most Delphi components work in design mode exactly the same way they work in execution mode, you can easily make your code detect whether or not you're still in design mode or execution mode. The flag `cdDesigning` in the `ComponentState` will give us this information, for example:

```
If csDesigning in
ComponentState then
  MessageDlg('Design Mode',
    mtInformation,[mbOk],0)
```

This very nice feature enables us to put a lot of extra (debug) code and information at design places that are no longer used then we generate a true .EXE. All with the same code!

QIs the file type Text no longer supported? I used it a lot in Borland Pascal, but this code gives me a compiler error Error 21: Error in type:

```
procedure TForm1.Button1Click(
  Sender: TObject);
var myFile: Text;
```

A The error message is produced because Text is also a property of the button. To get the Text file type, use `System.Text` to scope it to the system unit. The same is true for `Close` and `Assign`. Alternatively, use the new *Delphi only* `TextFile`, `AssignFile` and `CloseFile` procedures.

QI understand that identifiers in the private section of a class or object are of course private and identifiers in the public section are public, but what about the component declarations added automatically by Delphi *above* the private section. What access rights do they have?

A It seems that (unless specified otherwise) the fields, methods and properties of objects are all published: that is they are both public and automatically appear in the Object Inspector.

Published parts have run-time type information generated for them which is available to the Object Inspector. You can find more information on this subject in the *Delphi Component Writers Guide*.

QHow do I go about working with Microsoft Access files in Delphi? I've heard several people say they have had difficulties.

A You can work with Microsoft Access .MDB files in a Delphi application using an ODBC driver. Delphi actually gives you all you need, but the process is not immediately obvious! Here are the steps.

What You Need

First, check you have the ODBC Administrator installed (file `ODBCADM.EXE` in `WINDOWS\SYSTEM`, you also need the file `ODBCINST.DLL`, for installing new drivers, and `ODBC.DLL`). The ODBC Administrator should be shown in Control Panel as the ODBC icon. If you didn't have it already, it should have been installed when you installed Delphi.

If you get a message to the effect that "Your ODBC is not up-to-date. IDAPI needs ODBC greater than 2.0" you have an earlier version and should replace it with the one included with Delphi.

Check you have an Access ODBC driver installed in Windows. You can do this by clicking Drivers in the Data Sources dialog which appears when you run the ODBC Administrator. Delphi installs an entry Access Files (*.mdb) or Access Data (*.mdb), which works with Access **1.10** files and uses the `SIMBA.DLL` driver (note that this DLL also needs the files `RED110.DLL` and `SIMADMIN.DLL` - all installed for you by Delphi). These files are redistributable with your programs as part of the ReportSmith Runtime.

If you want to work with Access **2.0 or 2.5** files, you need to obtain a different set of driver files from Microsoft. The key file is `MSAJT200.DLL`, also needed are

MSJETERR.DLL and MSJETINT.DLL. Ask for the ODBC Desktop Drivers, Version 2.0. The cost in the USA is \$10.25. If you have MSDN Level 2 they are on the January Development Platform CD 4, in \ODBC\X86 with the ODBC 2.1 SDK.

There is apparently an update for these drivers for Access 2.5 files on the MSACCESS CompuServe forum.

Note that the Access ODBC driver included with certain Microsoft applications (eg MS Office) can only be used with other MS applications. They may appear to work but sooner or later will come back to bite you! The line to watch out for (and avoid!) in the ODBC Administrator drivers list is Access 2.0 for MS Office (*.mdb).

You can install new ODBC drivers using the ODBC Administrator in Control Panel.

Adding An ODBC Data Source

If you've got all the files you need, you are ready to proceed. The example presented here uses the Access 1.10 driver supplied with Delphi. Using the ODBC Administrator, set up a data source for your Access files:

- > Click Add in the data sources dialog box to display the Add Data Source dialog, then select Access Files (*.mdb) (or whatever the appropriate entry is for the driver you have installed).
- > In the ODBC Microsoft Access Setup dialog (Figure 1 top) enter a name in the Data Source Name. We'll use My Test. Enter a description of the Data Source in the Description text box.
- > Click Select Database to open the Select Database dialog. Navigate to the directory where your Access .MDB files are and select one (Figure 1 bottom).
- > We'll select a file TEST.MDB in a directory C:\DELPROJ\ACCESS. Click OK in the Setup dialog.

You will now have an entry in the list of Data Sources of My Test (Access Files *.mdb). Click Close to exit the ODBC Administrator. You can set up more Data Sources as required, using the same method.

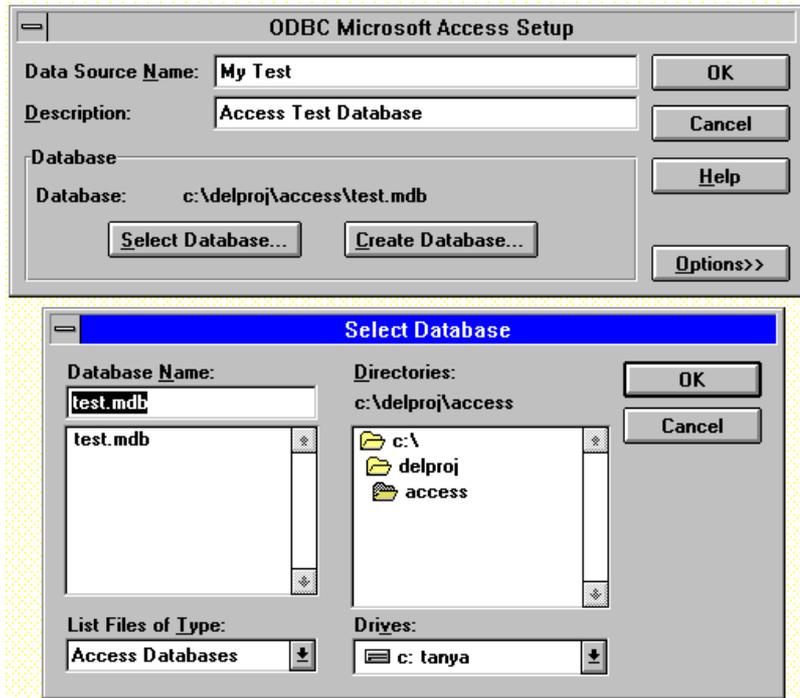


Figure 1



Figure 2



Figure 3

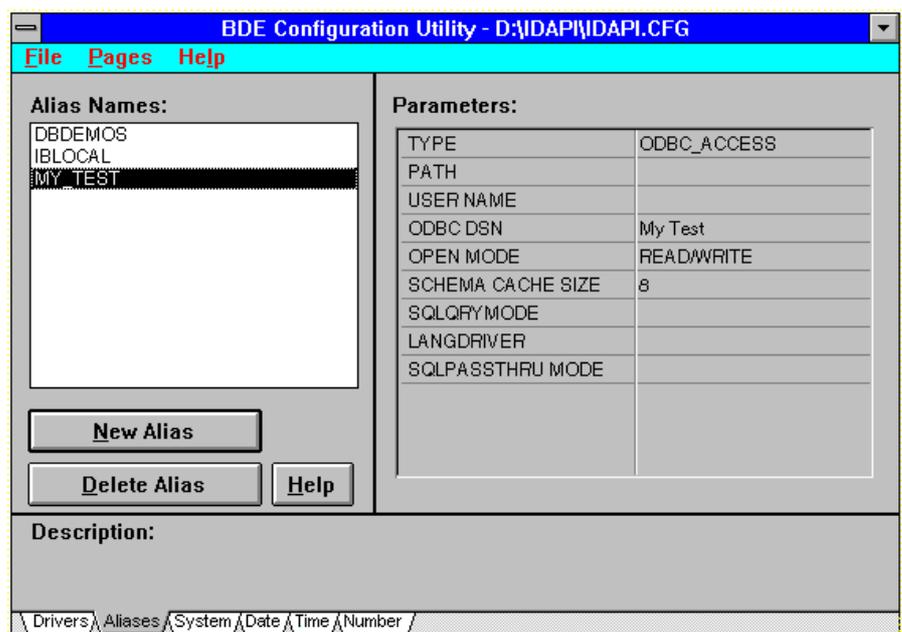


Figure 4

Setting Up The Borland Database Engine

Now load the Borland Database Engine (BDE) Configuration Utility. On the Drivers page click New ODBC Driver. Note that this adds an Access driver to BDE and is a completely separate exercise to adding an Access driver to Windows using the ODBC Administrator.

- > In the Add ODBC Driver dialog presented (Figure 2), enter ACCESS (or whatever you like) in the top edit box. BDE adds ODBC_ in front automatically.
- > In the combobox underneath select Access Files (*.mdb). Select the Data Source in the next combobox (Default Data Source Name) – these are the Data Sources you have set up in the ODBC Administration Utility. You needn't worry which one you select, as you can change it later as we'll see. Click OK.

Once you've set up this BDE Driver, you can use it to talk to more than one ODBC Data Source, using a different **Alias** for each ODBC Data Source. To set up an Alias, switch to the Aliases page and click New Alias.

- > In the Add New Alias dialog (Figure 3), enter an Alias Name of your choice. We'll use MY_TEST (note that spaces are not allowed).
- > In the Alias Type combobox, select the ODBC driver name that you just created (in our case ODBC_ACCESS). Click OK.
- > If you have more than one ODBC Data Source, alter the ODBC DSN Parameter ("DSN" = "Data Source Name") in the list of Parameters for the Alias to the appropriate ODBC Data Source (Figure 4), as set up in the ODBC Administrator.

- > Note that you needn't add anything in the Path Parameter, as the ODBC Data Source already has this information. If you **do** add a Path, make sure it's correct or things won't work!

Now save the BDE configuration by selecting File|Save on the menu bar, then exit the Database Engine Configuration Utility.

In Delphi

Start a new Project and drop a Table and DataSource onto the form from the Data Access page of the Component Palette. Then drop a DBGrid onto the form from the Data Controls page.

- > For the Table, set the DatabaseName in the Object Inspector to MY_TEST – the Alias you set up in the BDE Configuration Utility (Figure 5). Now go down and click the button in the TableName combobox.
- > You will be asked to Log In to the Access MY_TEST database. Note that you don't need a User Name or Password unless they have been specifically set up, so just click OK.
- > After a pause the list of available tables for the ODBC Data Source pointed to by the BDE Alias you have set up will appear in the combobox. Select TEST (Figure 5 again).
- > For the DataSource, set the DataSet property in the Object Inspector to Table1 (Figure 6).
- > For the DBGrid, set the DataSource property in the Object Inspector to DataSource1 (Figure 7).
- > Return to the Table and set the Active property to True in the Object Inspector.

The data from the TEST table will now be displayed in the grid. And that's all there is to it!

One thing to beware of is that if you create an application which uses Access tables and then Run it from within the Delphi IDE, you will get an error if you attempt to

modify the data in the table(s). If you run the compiled .EXE file outside of Delphi (having closed Delphi), everything will be fine.

The ODBC error messages unfortunately tend to be obscure and difficult to relate to what's going on in your application, but checking the setup in the ODBC Administrator and the BDE Configuration Utility usually sorts things out.

If you want more reference information, try the ODBC 2.0 Programmer's Reference and SDK Guide from Microsoft Press (ISBN 1-55615-658-8, US price \$24.95).

As we go to press, there have been various reports of problems with using ODBC for getting at Access files. Some of these reports have been from users of Delphi field test versions, where things didn't work fully. Other reports are simply a result of users not using the correct procedure to set up ODBC links – which is why we've included this run-down. It is possible that there may still be some genuine difficulties remaining and if so we'd like to hear from you!

Finally, note that if you need to create new Access 1.10 tables you can use the Database Desktop included with Delphi.

Helping with the answers for the problems in The Delphi Clinic for this issue were Ralph Friedman (CompuServe 100064,3102), Bob Swart and Chris Frizelle. Thanks everyone! Don't forget to keep sending in your problems and queries to us...

Figure 5

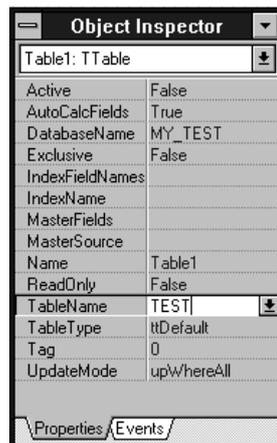


Figure 6

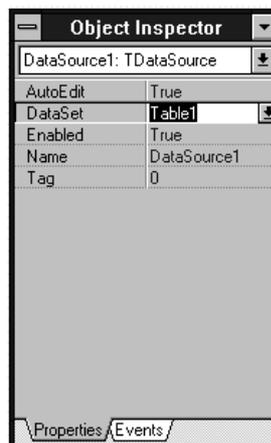


Figure 7

