The Hidden Power Of Jet

by Guy Smith-Ferrier

Jet is well known as the database engine for Microsoft Access databases. This article isn't, however, about Access databases. What is not so well known about Jet is the myriad of other formats which it can access. This article is about using Jet through ADO to access these other formats.

The Jet 3.51 OLE DB Provider first shipped with Microsoft Data Access Components (MDAC) in v1.5c, back in December 1997. It provided access to Access 97 databases. A year later, MDAC v2.1 shipped, including the Jet 4.0 OLE DB Provider. Jet 4.0 improved on previous versions by providing support for Access 2000 and giving access to installable ISAM drivers. It is these ISAM drivers which make Jet so powerful and so useful.

At the time of writing, the current release of MDAC is v2.6. This is the first release since v1.5c *not* to include Jet and it is clear that Jet will not be included in future releases of MDAC. Certainly, this is a problem if you read this article and see all of the clever tricks that the Jet engine can do and want to make use of them, but fear that you cannot redistribute Jet. So here's a quick rundown on how to get the Jet engine.

Firstly, if you have Windows 95, 98 or NT then you can download, distribute and install MDAC 2.5, which includes the Jet engine. If you have Windows 2000 or Windows ME, you already have the Jet engine, because they both ship with MDAC 2.5. Alternatively, your user can install any of the following products which include the Jet engine: Microsoft Office, Microsoft Excel, Microsoft Access, Microsoft Visual Basic or Microsoft Visual C++. Lastly, Microsoft have made mumblings about releasing the Jet engine separately as a free download on their website. At the time of writing, nothing has happened yet, but by the time you read this it might be worth checking again.

Paradox

One of the formats which Jet can access is our old friend and enemy, Paradox.

The approach to using Paradox files is exactly the same as using any of the formats which Jet supports. Add a `TADOTable` to a form and set `ConnectionString` to use the Jet 4.0 OLE DB Provider. If you were to follow the Microsoft Connection String Editor at this point then you would be entering the name of an Access MDB. Of course, this won't help you much if you want to use Paradox, so select the `All` page and set `Extended Properties` to `Paradox 7.x`. Now go back to the `Connection` page and enter the name of the directory containing the Paradox tables (eg, C:\program files\common files\borland\shared\data) in the database name. Done. Drop down the list of tables from `ADOTable1.TableName` and select a table (for example, `Customer`) and you are using Paradox.

At this point the eternal question 'Does ADO require the BDE for Paradox files?' usually rears its head.

If you haven't heard this question before you will probably be falling off your chair in hysterics at the idea that ADO needs the BDE to be installed in order to use Paradox. In terms of ironies this one's right up there. It would be really helpful if the answer to this question was either 'Yes' or 'No' but if it were that easy then there wouldn't be so much confusion. So here goes.

If you use Jet 3.51 then you don't need the BDE at all. If you use Jet 4.0 then you don't need the BDE if you are just reading the data in a single user environment. If you are using Jet 4.0 and you are updating the data or opening tables in a multiuser environment then you must also install the BDE. I'm sure you won't be surprised to hear that Microsoft received some criticism over this and eventually relented. You can now contact Microsoft Technical Support and ask for new Paradox ISAM drivers for Jet 4.0 which do not require the BDE. Sadly, at the time of writing, they are not available for public download.

Excel

Now that we have started the ball rolling with Paradox, let's see what else we can get at. Jet is installed with Excel, so it isn't surprising that the Jet OLE DB Provider can read and write Excel spreadsheets.

For Excel you will need to use a `TADODataSet` because the `SELECT` statement is not formulated correctly for Excel when you use a `TADOTable`. Add a `TADODataSet` to a form and set the `ConnectionString` to use the Jet 4.0 OLE DB Provider. As before, you need to select the `All` page and set `Extended Properties` to `Excel 8.0`. In the `Connection` page set the data source to the .XLS filename. The `TADODataSet` component uses the information in the `CommandText` property to open a data source. You tell the `TADODataSet` how to interpret this property using the `CommandType` property. Set `CommandType` to `cmdTableDirect` to indicate that the `CommandText` is the name of a table. Now click on the `CommandText` property, drop down the list, and you will see the list of workbooks in the .XLS file. The names will all be suffixed with a dollar sign, so if the workbook name is `Contacts` then the table name will be `Contacts$`. Open the dataset, add a `TDataSource` and a `TDBGrid`, and you will then be viewing an Excel spreadsheet.

You will notice straight away that every column is of width 255. To get around this, you can set the column widths by adding persistent fields and setting their widths, or you can add columns to the grid and set their widths.

If you run the program from the IDE you will discover that the Excel

Format	Extended Properties
dBASE	dBASE III, dBASE IV, dBASE 5.0
Paradox	Paradox 3.x, Paradox 4.x, Paradox 5.x, Paradox 7.x
Excel	Excel 3.0, Excel 4.0, Excel 5.0, Excel 97, Excel 8.0
HTML	HTML Import, HTML Export
Text	Text

➤ *Table 1*

ISAM driver opens the Excel file exclusively and your application will crash, because the IDE has already opened the Excel file. To avoid this, close the application in the IDE and run it from Windows Explorer. The good news is that you can even edit the data in the Excel file and add new rows. The bad news is that you cannot delete rows.

Remember that all that the driver has done is open the file: it is not running Excel. This means that calculations are not recalculated when the data is changed, for example.

Extended Properties

So now we have done a couple of examples and you have seen that both solutions involved setting the `Extended Properties` dynamic property. You are probably wondering what other values it can be set to. Table 1 shows the most commonly available ISAM drivers.

The exact list is dependent upon what is installed on your machine but you can find this out by looking in the registry at the following key value:

```
HKEY_LOCAL_MACHINE\Software\
  Microsoft\Jet\4.0\
  ISAM Formats
```

Text

Just as the BDE allows you to open text files using its `ASCIIDRV` driver so ADO supports access to text files through the Jet OLE DB Provider. This time around you need to set `Extended Properties` to `Text` and the data source to the directory containing the text file or files. Start with a simple file like the one below (we'll call it AREAS.TXT):

```
Codes    ,Names
SY       ,Surrey
KT       ,Kent
SG       ,South Glos.
```

You can use any ADO dataset, but `TADOTable` is the simplest to use here. Set the `ConnectionString` and then drop down the `TableName` list. All of the text files will be listed as table names. The dot before the file extension will have been translated to a hash sign (ie `AREAS#TXT`). Set `Active` to `True` and add a grid to view it and you're all done. Unfortunately, you can't edit or delete records, but you can add them. As in Excel the field sizes default to 255. To overcome this, you can either make the adjustments in Delphi or provide the Text ISAM driver with a bit more information. This is where SCHEMA.INI comes in.

SCHEMA.INI is the Text ISAM's means by which you can supply more information about the text file. Without SCHEMA.INI the driver has to make a few guesses. For example, it has to guess whether the first row contains field names or data, the data types of each column, the format of the text file (eg CSV, tab-delimited, etc). In reality the guesses are quite good, because it takes a peek at the text file it is opening. By default it reads the first 25 rows and tries to decide what format the file is in, based on what it finds. Also, without the aid of SCHEMA.INI, it looks in

```
HKEY_LOCAL_MACHINE\Software\
  Microsoft\Jet\4.0\Engines\
  Text
```

for other default settings for the driver.

So what is SCHEMA.INI? It is an optional .INI file placed in the same directory as the text files to which it refers. It contains additional information about those text files. Below is a simple SCHEMA.INI which can be used with the previous AREAS.TXT file:

```
[areas.txt]
ColNameHeader=True
Format=CSVDelimited
Col1=Codes Char Width 2
Col2=Names Char Width 30
```

The `ColNameHeader` line indicates that the first line contains column names and not data. The `Format` line indicates the format, which can be `CSVDelimited`, `TabDelimited`, `CustomDelimited` (where you can specify your own delimiter character or characters) or `FixedLength`. The `Col` lines specify the format of the columns. This is particularly useful for clarifying tricky date formats. The number of options available in SCHEMA.INI is considerable (search on SCHEMA.INI in the Platform SDK for more information) and puts `ASCIIDRV` in the Borland Database Engine to shame.

Exporting Data

Another useful feature of the Jet OLE DB Provider is the ability to export data using any ISAM driver.

For example, you can export an Access table directly to Excel with just a `TADOConnection`. Add a `TADOConnection` to a form, set its `ConnectionString` to use the Northwind.mdb database via the Jet 4.0 OLE DB Provider and set `Connected` to `True`. Add another button, with just one line of code, as shown in Listing 1, and your data will be translated as if by magic.

The `SELECT` statement is the clever part and uses a couple of non-ANSI standard extensions to get the job done. The essential `SELECT` statement is:

```
SELECT * FROM CUSTOMERS
```

➤ *Listing 1*

```
ADOConnection1.Execute('SELECT * INTO CustomersSheet IN ' +
  '"C:\temp\NorthWind.xls" "Excel 8.0;" FROM Customers');
```

but as you can see there are other parts to it. The INTO clause specifies the new table in the destination database.

In Excel terms, a table is a worksheet and in this example it is called CustomersSheet. The table must not exist already. The IN clause specifies the database to which the table will be added. Again, in Excel terms, this is an Excel .XLS file and in this example it is C:\temp\Northwind.xls. If the database exists then it is opened and the table is added. If the database does not exist then it is created. Note that this is the simplest way to create an Excel spreadsheet in ADO.

The next piece (Excel 8.0;) is the ISAM driver used to export the data to. There is a huge gotcha in this part of the statement: the trailing semi-colon is *not* optional.

The syntax is not limited to using just the Excel ISAM driver. Any export driver can be used (in practice this means any driver except HTML Import). So we can take what we have learnt and apply it to using the HTML Export ISAM driver. Add another button to the form, with the code from Listing 2.

This example exports the Customers table to an HTML document. The table name is [NorthwindCustomers.htm] (the square brackets are not optional). The database name is a directory. When you execute this line the HTM file is created and also a SCHEMA.INI file (see Listing 3).

HTML Import

The last of the ISAM drivers which has a special significance is the HTML Import driver. As the name implies, this is for importing HTML into an ADO recordset. In essence, it allows you to open an HTML table and treat it like a database table.

Add a TADOTable and set the ConnectionString to use the Jet 4.0 OLE DB Provider and set Extended Properties to HTML Import. Set the data source to C:\Temp\NorthwindCustomers.htm (that is, the file which was created by the previous export). This appears to be a curious contradiction. Using the HTML Export driver the database name was the name of the directory but using the HTML Import driver it is the name of the file. Let's carry on and pretend we haven't noticed. Set the TADOTable's TableName to NorthwindCustomers. This is the CAPTION tag of the HTML table. Set Active to True and add a TDataSource and a TDBGrid to view the results. You are now looking at an HTML table in a TDBGrid. Clever stuff.

```
ADOConnection1.Execute('SELECT * INTO [NorthwindCustomers.htm] IN ' +
  '"c:\temp" "HTML Export;" FROM Customers');
```

➤ *Listing 2*

Run the program and you will see that you cannot insert, edit or delete data. Of course, the reason for this lies in the name of the driver, HTML Import, which shows that it is not designed for updating the HTML.

One disappointing observation on the TADOTable is that all of the fields are of length 255. This is disappointing because, as we saw when we exported the data, there is a perfectly good SCHEMA.INI file in the same directory, showing all of the field sizes.

Conclusion

So what do we learn from all this? We learn that the Jet OLE DB Provider is a lot more versatile than it is given credit for. Certainly it is used for accessing Access databases, but it can also be used to access a wide variety of desktop databases via installable ISAM drivers.

Of course, its future looks a little doubtful with its removal from MDAC, but at present you can still install the Jet engine on your users' machines simply by installing MDAC 2.5 and then upgrading them to a more recent MDAC release (if needed). As both Windows 2000 and Windows ME come with MDAC 2.5 this is all still viable.

```
[NorthwindCustomers.htm]
ColNameHeader=True
CharacterSet=1252
Format=HTML
Col1=CustomerID Char Width 5
Col2=CompanyName Char Width 40
Col3=ContactName Char Width 30
Col4=ContactTitle Char Width 30
Col5=Address Char Width 60
Col6=City Char Width 15
Col7=Region Char Width 15
Col8=PostalCode Char Width 10
Col9=Country Char Width 15
Col10=Phone Char Width 24
Col11=Fax Char Width 24
```

➤ *Listing 3*

The problems will come when a new version of Windows ships with MDAC 2.6 or later. However, there is still the possibility that Microsoft will fulfil their promises of making the Jet engine available for download independently of MDAC. We shall have to wait and see, but here's hoping.

Guy Smith-Ferrier is a Senior Delphi Consultant for Inprise Professional Services Organisation in the UK. He continues his ambition to play the piano better than a deaf one-armed monkey but is beginning to realise that he has met his match. Contact him at guysmith@compuserve.com