

Using Microsoft 2007 Drivers (ACE.12 and ODBC) in Winbatch

With Office 2007, Microsoft not only changed the interface (replacing the drop-down menus with the 'ribbon') but added new file formats and extensions to the Excel and Access applications. For compatibility with Office 2003 and for use with existing OLE applications, Microsoft released both an Office Compatibility Pack (for Excel and Word so you can view 2007 documents in 2003) and a driver set consisting of both an OLEDB Provider (specific to Access 2007) and an ODBC driver (for both Excel and Access in both 2007 and earlier formats).

I have tested the new drivers and found that they do not completely correspond to previous scripts I had written. Not a huge issue, some small frustrations at first, but perhaps what follows will save others a few critical error runs. The example pertains to Access but, at least for the ODBC driver, applies to Excel.

The example assumes you wish to open any Access Database and export table information to an XML persisted database. To perform this you need to

- Select the file
- Open an ADO Connection
- Issue an OpenSchema(20)
- Persist the resultant recordset as XML

If you use an OLEDB Provider, you must differentiate by the file extension (.mdb or .accdb).

```
types="Access 2007/200-2003 | *.accdb;*.mdb | "
```

```
cMDB=AskFilename("Access File To Enumerate", dirsript(), types, "", 1)
```

```
If cMDB="" Then Exit
```

```
ext = FileExtension(cMDB)
```

```
cXML = StrReplace(cMDB,ext,"xml")
```

```
If FileExist(cXML) Then FileDelete(cXML)
```

```
If StrUpper(ext)="MDB"
```

```
    cProv="MicroSoft.Jet.OLEDB.4.0"
```

```
Else
```

```
    cProv="Microsoft.ACE.OLEDB.12.0"
```

```
Endif
```

You then set up your connection string as

```
cConn = "Provider=:cProv:;Data Source=:cMDB:;" (using WB 2007 concatenation)
```

and then proceed with

```
oConn = CreateObject("ADODB.Connection")
```

```
oRS=oConn.OpenSchema(20)
```

```
message("Records",oRS.RecordCount)
```

```
oRS.Save(cXML,1)
```

```
oRS.Close()
```

I threw the message() statement in as it is important to comparing the Provider and the Driver, but up until now we have a pretty standard WB OLE script that uses ADO to get at a recordsource, as under ADO, `oRS=oConn.OpenSchema(20)` opens an 'implicit' Recordset Object without it needing to be declared.

Using the ODBC Driver, does not require extracting the file extension as it handles both file formats:

```
cConn = "Provider=MSDASQL;Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=":cMDB:";
```

This will set up access to the DB so that the rest of the script can follow the same ADO (which is cool and generic). What is more significant is that Microsoft is deprecating the MSDASQL Provider and you can just as easily use

```
cConn = "Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=":cMDB:";
```

NOTE: the MSDASQL Provider is still useful for generic ADO for drivers like MySQL, SQLite, Cheetah etc...

So, at this point it seems a simple choice – Provider or Driver. Assume that the database you chose to examine had 21 tables (both regular and system tables). If your were using the ODBC Driver, the message() line would return 21, if using the Provider it would return -1. If using the Provider the oRS.Save() would work, the ODBC driver would error with 'Unspecified Provider Error'.

Like I said, initial frustration for me... but once I got over the Evil Microsoft feelings it just made sense that they were moving to a more strict typing environment, something you see when comparing VSTO to VBA.

So, simple fix: declare an explicit Recordset and explicitly declare both the Connection and Recordset as adUseClient [3]:

```
oConn = CreateObject("ADODB.Connection")
```

```
oConn.CursorLocation = 3
```

```
oRS = CreateObject("ADODB.Recordset")
```

```
oRS.CursorLocation = 3
```

```
oConn.Open(cConn)
```

```
oRS=oConn.OpenSchema(20)
```

Stan Littlefield, August 25, 2007