

Re: Creating dbf table using ADO Recordset open method

Source: <http://www.tech-archive.net/Archive/Data/microsoft.public.data.ado/2006-09/msg00161.html>

- *From:* "Ralph" <nt_consulting64@xxxxxxxxxx>
 - *Date:* Wed, 27 Sep 2006 21:21:43 -0500
-

<vp75040@xxxxxxxxxxxxxxxxxx> wrote in message
news:cqmlh2lhoqsv0evkqt47qkp98l8u6gl6u5@xxxxxxxxxx

Hi All,

I have an MFC applicaiton where I connect to a dbf file using ado and then get a recorset from this file. All works fine. What I then do is I create another dbf table by using CREATE Table ddl statement in the recordset open method. It creates a table just fine, however if the table name specified is longer than 8 characters is will truncate to 8 characters. Anyway around this? I am importing msado15.dll into my MFC app.

Victor

If I remember correctly dBase limited field names and table names to 8 characters (first can't be a number). Table names are associated with File Names so they were limited to the old Dos 8.3 format.

Since then other version of "xBase" have appeared, I think names were extended to 24 and 32 characters. Even here you have to be careful because (like past C compilers) the tool would accept a 32-char name, but only the first 24 were significant.

It also depends on the tool you plan to use the .dbf file – obviously if it can't accept 32-char 'names' – it does you little good to create them. The provider/engine you are using now apparently is honoring only the least common denominator.

You mentioned using msado15.dll, but that doesn't provide any clue as to what version of ADO you are using, or the provider. M\$ has been spotting on its "xBase" support. (At one time removing it from the MDAC/Jet packages)

However, all is not lost. You may be able to use the FoxPro provider and create xBase files which will allow longer names. (The core 'table' files

Re: Creating dbf table using ADO Recordset open method

for FoxPro and xBase). Also the format for a dBase file is very simple and well published. (A little googling should find numerous sites.) Using C++ you should be able to whip your own writer out in less than a day. (Again you would have to make sure the receiving tool can handle longer names.)

There are also a ton of source code and several 3rd party tools you can employ.

hth
-ralph