Re: adodb.recordset.save and back

Source: http://www.tech−archive.net/Archive/Data/microsoft.public.data.ado/2004−03/0029.html

---

From: Rinoo Rajesh (*rinoorajesh_at_yahoo.com*)
Date: 03/02/04

Date: Tue, 2 Mar 2004 22:32:15 +0530

What I can gather from your mail trail below is that you wish to persist a recordset into XML (and you have been successful in that) and now wish to have a method to update back the data into some table − by loading the persisted XML file back into recordset, right?

Well, if you wish to do what i have outlined in the text above, there is no straight way, but it is possible. You see, the problem is that it is easy to load the XML file back into the recordset (would be a disconnected one at that moment anyway) − for example:

Dim Rs as New ADODB.Recordset

RS.Open "XML FILE PATH"

As easy as that!

However, the problem starts when you wish to use the RS.Update or RS. UpdateBatch stuff. ADO will throw out an error stating that there is insufficient base table information to perform an update. you can't then connect to a database and perform
any batch updates. The problem is that a stand−alone Recordset has an empty Source property, and when you reconnect it to the database you get the error "Insufficient Base Table Information".

What is missing in a stand−alone Recordset are three field attributes that indicate from which database, table and field the
corrisponding value comes. If you create a new Field object using the Fields.Append method you can't specify these properties, nor you can add them later because field properties can't be extended.

This can be handled, via XML Schema that you have embedded in the XML persisted recordset. At this point, I would like to draw your attention to an excellent article on VB2TheMAX titled "Connect a stand−alone Recordset to a database using XML". Search it on the google if you can't locate the article. It provides sample code and technique to do the stuff.

Let me know if you need more help on this one.

Regards,

Rinoo Rajesh

"Peter Plumber" <klempner@gmxdot.net> wrote in message
news:eeavaot#DHA.624@TK2MSFTNGP11.phx.gbl...
> *Hi again,*
>
> *could anyone please at least tell me that it is no magic function for:*
>
> *Dim cn As New ADODB.Connection*
> *Dim rs As New ADODB.Recordset*
>
> *cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data*
> *Source=db1.mdb"*
> *cn.Open*
>
> *rs.Open Source:="tblUser.xml", Options:=adCmdFile*
>
> *magic_function rs, cn ' writes the rs to the datasource opened with cn*
>
> *rs.Close*
> *cn.Close*
>
>
> *this is part of my "Not_So_Magic_Function":*
>
> *Do Until rsSrc.EOF*
> *rsDst.AddNew*
>
> *For Each fld In rsSrc.Fields*
> *rsDst.Fields(fld.Name).Value = fld.Value*
> *Next fld*
>
> *rsDst.UpdateBatch adAffectCurrent*
>
> *rsSrc.MoveNext*
> *Loop*
>
> *this works but is rather slow compared to*
> *rs.Save "tblUser.xml", adPersistXML*
>
> *thx*
>
> *Peter*
>
>
> *"Peter Plumber" <Klempner@gmxdot.net> schrieb im Newsbeitrag*
> *news:eGc15yY#DHA.2524@TK2MSFTNGP11.phx.gbl...*
> *> This is what I have:*
> *>*

> > *Dim cn As New ADODB.Connection*
> > *Dim rs As New ADODB.Recordset*
> >
> > *cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data*
> > *Source=db1.mdb"*
> > *cn.Open*
> > *rs.Open "SELECT * FROM tblData", cn*
> > *rs.Save "tblUser.xml", adPersistXML*
> > *rs.Close*
> > *cn.Close*
> >
> > *and a very nice "tblUser.xml"−file with all the data from tblData*
> >
> > *is there a fast way to get the data back from the xml−file into the*
table
>
>