

Re: Verification of replication

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.replication/2005-07/msg00008.html>

- *From:* MWG <MWG@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 5 Jul 2005 21:23:02 -0700
-

Well, I've obviously been very lucky! I have been convinced to change my habits.

Thanks for the input...One and all!

"David W. Fenton" wrote:

> =?Utf-8?B?TVdH?= <MWG@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in
> news:B6E42DAD-B6C2-4B14-B08F-87E663AF36C7@xxxxxxxxxxxxxx:

>
> [quoting me:]
>> Bzzzt. They shouldn't be sharing the front end. This can lead to
>> major problems in that front end, leading to corruption of it,
>> and even to corruption of the back end data file.

>>
>> First, this project is still in testing so the users are working
>> on shared f/e to "break" it. However, even if they weren't
>> testing, I can see this happening if the users were changing any
>> of the objects. But what you're saying is that it is coincidental
>> that I've not had any major corruptions over the last 6-7 years
>> using the shared f/e environment? . . .

>
> Well, with Access97 you could often get away with it, but A2K
> changed all of that. If you haven't had any problems, then you've
> been extremely lucky.

>
>> . . . Speed issues, yes...but db
>> corruptions, not really. So, you really think I should maintain
>> an mdb f/e for dev purposes and create/distribute an mde f/e to
>> each client machine (a suggestion from one of the articles you
>> sent me to)? . . .

>
> Absolutely. I have never once in the nearly 10 years I've been doing
> Access development for a living done anything else.

>
>> . . . Is that truly better in an environment of no more
>> than 3 simultaneous users adding/editing different data? . . .

>

Re: Verification of replication

> It's better in *all* environments, even with a single user.
>
>> . . . (I want
>> to see the good in this...I'm just not convinced yet – or maybe
>> I'm just not understanding it totally). P.S. – Sorry...I know this
>> isn't a replication issue...but I didn't think I was asking for
>> help on it. It was simply given for scenario setup. Anyway,
>> thanks for the free tip.
>
> Everyone I know of who makes a living with Access development splits
> the app and puts a front end on every workstation. It saves all
> kinds of headaches.
>
> There are a whole host of methods for pushing front end updates out
> to the users.
>
> []
>
>>> . . . And if so, can anyone offer
>>> some help on the coding of the synchronization (I don't want to
>>> make the users go through the interface) as well as refresh my
>>> memory on the order in which the replicas need to be built. (I
>>> know this can get screwed up pretty easily and I'm limited on
>>> how many "do-overs" I get).
>>>
>>> It's pretty simple:
>>>
>>> Public Sub SynchReplica(dbSource As DAO.Database, _
>>> strDestinationMDB As String, _
>>> Optional StrDescription As String = "Replica")
>>> Dim strMsg As String
>>>
>>> If Len(Dir(strDestinationDB)) <> 0 Then
>>> If Len(Dir(strDestinationMDB)) = 0 Then
>>> 'create replica
>>> dbSource.MakeReplica strDestinationMDB, StrDescription
>>> Else
>>> 'synch with replica
>>> dbSource.Synchronize strDestinationMDB
>>> End If
>>> Else
>>> strMsg = "Could not connect to " & strDestinationDB
>>> strMsg = strMsg & vbCrLf & " " & vbCrLf
>>> strMsg = strMsg & "You may not be connected to the network."
>>> MsgBox strMsg, vbExclamation, "Could not find replica!"
>>> End If
>>> End Sub
>>>
>>> You might want to add an error handler in the event that
>>> something else goes wrong. You also might want to check for
>>> conflicts/errors. I posted code recently about that. You should

Re: Verification of replication

>>> be able to hit Google Groups for this newsgroup and search for
>>> posts under this email address and the word "conflicts" in the
>>> body.
>>>
>>> You'd call the code above like this:
>>>
>>> Dim db As DAO.Database
>>>
>>> Set db = DBEngine(0).OpenDatabase([path/name of your back end])
>>> SynchReplica db, [destination replica path/name], [description]
>>> db.Close
>>> Set db = Nothing
>>>
>>> That could easily be the code behind a command button, or if
>>> you're using the Switchboard manager to create a switchboard, you
>>> could put the code above inside a function in a global module,
>>> and call the function from the switchboard manager.
>>>
>>> So, you see, it's really not complicated at all.
>>
>> This is excellent! And you're right...not all that tough. You've
>> been very helpful...thanks so much!
>
> Well, I was just copying my own code -- pretty easy for me to do!
>
> I actually chose the simplest example (well, really I altered a
> backup routine), because most of my actual code for synching is more
> complicated, since I check for conflicts/errors. I didn't want to
> confuse the issue.
>
> []
>
>>>> Everything I've tried negates the idea of replication and I
>>>> simply don't have time to code my own synch process. The only
>>>> viable thought I've had was to apply a filter to the forms to
>>>> only display the chosen case(s). That way the data stays
>>>> intact for replication when the NB gets back home.
>>>>
>>> Well, do you know who the user of the database is? Do they log
>>> onto the laptops with a user name that you can use to filter?
>>
>> It's based less on the user and more on the cases. I am logging
>> who is doing what, when so I can track them down if need be. My
>> thought was to ask the user for the cases then filter the forms to
>> only show those cases (they want me to remove the other data all
>> together – but if I just "hide" it, my replica sets will be
>> complete and I'll have fewer conflicts, don't you think?)
>
> Well, that's basically what I outlined below.
>
>>> One way is to ask who they are at logon and filter accordingly,

Re: Verification of replication

> >> but that's easily broken.
> >>
> >> The easiest way is to just trust them to really only give a rat's
> >> ass about their own data, and filter the form after it opens.
> >
> > They care...they're betting on the NBs getting stolen and
> > compromising info – that's why they don't even want to take it on
> > the road with them. I just may not be the one able to make that
> > happen for them.
>
> Well, that's an issue of making sure the notebooks are properly
> secured. I assume they are Win2K or WinXP. Here's some things that
> should/could be done to secure them (BTW, none of this is an Access
> issue, because Access can't be secured from someone who can log onto
> the network/laptop where the data file is located):
>
> 1. make sure the user logon is a non-obvious username, and a strong
> password (i.e., 8 characters or longer, mixing upper/lower case, as
> well as numbers and letters and non-alphanumeric characters, like !
> # @ \$ % | and so forth).
>
> 2. make sure that the users don't log on as an administrator. In
> fact, you may choose to disable the administrator account on the
> laptop, and allow it to be administered only when connected to the
> office LAN by authenticating to your domain controller as a domain
> administrator. However, this means that if something goes wrong in
> the field, nobody will be able to log on and administer it from
> there.
>
> 3. make sure it doesn't cache domain logons. That's a system policy,
> and the default is to cache 10 of them. That allows a user to log on
> as they would on the LAN without being connected to the LAN. It's a
> very bad thing for security of travelling laptops, though. When
> travelling they really need to log onto the local machine only.
>
> 4. naturally, any laptop should never be set up to automatically log
> on, or with no passwords for the users.
>
> There are some other more complex things that can be done, but your
> IT support people should be up to speed on those.
>
> Access isn't really securable (it's too easy to Google and find
> utilities to crack Jet user-level security), so just get used to
> depending on securing the laptop.
>
> >>> One way to do this would be to have the forms open with a one-row
> >>> blank recordset. You can do this with SQL like this:
> >>>
> >>> SELECT TOP 1 Null As FirstField, Null As SecondField,
> >>> Null As ThirdField [etc.]
> >>> FROM tblMainTable

Re: Verification of replication

Re: Verification of replication

> >>
> >> Since you're returning TOP 1, there's only one record, and since
> >> you've replaced all the field values with aliases for NULL, it's
> >> uneditable.
> >>
> >> Then have a combo box on the form that loads the chosen case.
> >>
> >> That's the design in most of my apps. It displays a minimum of
> >> data at a time, and is efficient.
> >
> > Now this...is an excellent idea. Thanks again for your input! It
> > is truly appreciated!
>
> That's my standard design. It's both efficient from a network
> standpoint, and also limits the amount of data that someone can very
> easily steal with copying to the clipboard.
>
> --
> David W. Fenton <http://www.bway.net/~dfenton>
> dfenton at bway dot net <http://www.bway.net/~dfassoc>
>
>

• *References:*

- ◆ ***Re: Verification of replication***
◇ *From:* David W. Fenton

- Prev by Date: ***Re: Replication for Access***
- Next by Date: ***Re: Replication for Access***
- Previous by thread: ***Re: Verification of replication***
- Next by thread: ***Creating a unique key conflict resolver***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***