

Re: Replication/Synchronization and security

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.replication/2005-06/msg00027.html>

- *From:* "David W. Fenton" <dXXXfenton@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 14 Jun 2005 00:59:47 GMT
-

"=?Utf-8?B?QWxwYWNpYW4=?=" <Alpacian@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in news:078504ED-EA15-494C-85D9-B2CFF1E3809C@xxxxxxxxxxxxxxxxxx:

- > I have a situation with front end and back end databases that
- > require security because it contains medical data. Because these
- > databases need to be used by more than one person it resides on a
- > network with folder level security and the network has a very slow
- > speed for these users.
- >
- > My thinking is to use replication.

No.

Replication is not for solving application performance issues.

Secondly, by replicating the data to all the workstations you've compromised security -- you've multiplied the number of locations where data could be compromised.

Figure out why the application is slow and fix that.

Free hint: it's not because of security on the file share where the back end is stored.

It's most likely a design problem with your application.

- > I would like to create replicas of the back on the users machine
- > (C drive) that is synchronized to the network file when it opens
- > so they have the latest information and then synchronizes when
- > they close it to have a central storage place for all the users.
- > Because of the way our network is configured I would then like the
- > data on the C drive to be deleted when they are not using the
- > database to decrease the potential for breach.

No, you can't do that, as this means you are creating dead replicas. That will eventually lead to corruption of your replica set, eventually possibly to the point of complete loss of the full replica set and all its data.

Re: Replication/Synchronization and security

- > I have looked through the posts and was not able to find one that
- > answered my question.

You have the wrong idea. It's fundamentally wrong in so many ways as to almost defy refutation.

- > I have gotten as far as creating the secured replica of the back
- > end and it does sync but need to get the front end to sync the
- > back end as the user begins to use the database (using the
- > jrSyncTypeImport option or the blank database on the users machine
- > will wipe the master—I think)

I think JRO is a waste of time, especially when you're on a LAN and can use native DAO methods to synch.

One main concern:

How will you handle conflicts? That is, what if one person edits a record on their workstation and somebody edits the same record on a different workstation, what happens when those two data files are synched?

Your problem is application design.

If the application is running slowly over the LAN, then you need to redesign it to be more efficient. Here are some things to do:

1. never load more records than the user is going to use. That means not binding forms to tables, but to a SQL string that is set by the user requesting a record. If it's medical data, you probably have some form of patient id number and that can be the basis for the data retrieval. This also enhances security, since it makes it harder for somebody to copy large blocks of data to the clipboard and paste into Excel (which is possible when you've loaded large numbers of records in a form).
2. for dropdown picklists with large numbers of records in the base lookup table (leaving aside that it's almost always a bad idea to use such controls for more than a handful of records), don't populate the rowsource of the dropdown until the user has typed a few characters, and then filter it by what they've typed. You do this in the OnChange event of the combo box. You check the length of what's entered and when there's 2 or 3 characters, you then use that in a WHERE clause on the SQL string that sets the rowsource for the combo box.
3. never assign a recordsource or rowsource to a subform or control that is not onscreen. This means that if you have a form with multiple tab pages and a subform on each page, you load the subform only when that tab is selected, using the Tab's OnChange event.

Re: Replication/Synchronization and security

4. for seldom-changing lookup lists, you can copy the data over to the front end when the user logs on. This means that to get that data for dropdown lists and the like, the workstation doesn't have to make but the one request across the LAN. However, this is a very small performance improvement, as that kind of data is usually cached after retrieval, anyway.

Last of all, consider the nature of the network. It could be screwed up. It should be 100BaseT these days -- I don't even know if you can buy 10BaseT NICs any more. And you shouldn't even think about using Access about a wireless link, as this will unquestionably lead to corruption of your back end database with the first dropped connection during a data edit.

Also, if there is something wrong with the network that's leading to performance problems, then it's likely to mean the the network is just not reliable for data transfer, which means you are possibly exposed to corruption of the data.

Another solution that some would recommend is to switch the back end to SQL Server, MSDE, MySQL or some such, but if you're having performance problems because of application design, you'll just be pushing all those performance bottlenecks onto an overworked database server. You could see some improvement if the slowdowns are due to saturation of the network (with a database server less data is going to be pulled across the wire, though not nearly as much less as you might think, or as much less as many Access haters will wrongly tell you), but that's very unlikely to be the cause.

In short, converting the back end to a server database doesn't usually improve performance massively unless your application is already designed to be efficient, and if it were, you wouldn't be experiencing problems in the first place!

But, very definitely, replication is absolutely NOT the answer, especially in the scenario you've outlined with deleting replicas, which is an absolutely disastrous action to take.

--

David W. Fenton <http://www.bway.net/~dfenton>
dfenton at bway dot net <http://www.bway.net/~dfassoc>

-
- *Follow-Ups:*
 - ◆ **Re: Replication/Synchronization and security**
◇ From: Alpacian
 - *References:*

Re: Replication/Synchronization and security

◆ **Replication/Synchronization and security**

◇ *From:* Alpacian

- Prev by Date: **Replication/Synchronization and security**
- Next by Date: **Re: Replication/Synchronization and security**
- Previous by thread: **Replication/Synchronization and security**
- Next by thread: **Re: Replication/Synchronization and security**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**