Re: ADO Connection Timeout

Source: http://www.tech−archive.net/Archive/Data/microsoft.public.data.ado/2007−01/msg00005.html

- *From*: "Mark McGinty" <mmcginty@xxxxxxxxxxxxxxx>
- *Date*: Tue, 2 Jan 2007 11:47:07 −0800

Not that I want to beat you with a dead horse :−) but... for the sake of argument, more comments inline...

"Robert McCarter" <RobertMcCarter@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:39281A7F−6A44−4F0D−A197−6F513342EF14@xxxxxxxxxxxxxxx

> Hello,
>
> Sorry I didn't get into the particulars of the application. I have 4 test booths that are running tests on equipment. Each test booth runs a unique test. The stands are linked together on the line in a 1, 2, 3, 4 fashion.
> When the first test is run, the results are stored in the central database.
> When the equipment gets to the second stand, it needs to make sure the the first test was successful. If not, there is no need to run subsequent tests.
> This process continues through all four stands.

Ok... so what happens when a connection failure forces one station to revert to a local database? If your data acquisition process is also responsible for the failover, is it not halted while that takes place? How does it know when to try to reconnect to the central store again? And how much time does that take?

Also, what is the time frame for each set of tests, and transit between stations?

If the second process sent its data block after each tested item was complete, it would be just as much there as if you did it all from one process −− difference being that the first process would be relieved of the responsibility [and time committment] for insuring a working connection, and i/o delays caused by contention would be out of its critical path. With only one process responsible for all, every single glitch in connectivity represents a disruption, correct?

Just out of curiosity. are you saying the downstream test stations are fed by only one station in the first position? So if an item fails a test at

any station, the stations after it are idle? (These pertain more to the design of the line than that of your code, I realize.) I'd hope that each station's test set takes longer than the one before it, otherwise later stations will have built−in idle time by design.

> At each stand, when the equipment arrives, I have to pull in specification data as well as result data from the central database for each piece of equipment. I realize that it is extremely impossible for you to know what my
> application is doing without seeing it in the production environment. This
> is why in my original thread I was only looking for a way to control the timeout of the ADO connections. But as I am finding, there really isn't any
> way to set this timeout to a shorter length.

I think the fact that connection timeouts are a problem at all here is symtomatic of environmental issues. It sounds like these stations are in close enough proximity so as to be on the same network segment (no routers in between) and if you can't keep a stable connection there, you've got problems. Is it a full−duplex network? Is it heavily used? Might want to run NETSTAT to check collision counts.

> I know there is much better technology available for a project like this. As I replied to Mark, I would much rather use SQL Server and may get that opportunity later.

You realize of course that MSDE and SQL 2005 Lite are freely redistributable.

> This project as it stands though, must be reliable and not crash.

Therein lies the problem: you are trying to do something that's border−line impossible. Just as sure as I am sitting here typing from atop my little soapbox, that Jet database shared 5 ways *will* crash, all stations will be disconnected, someone will have to repair it, and it will quite likely lose some data. And that's on top of the connection problems you're seeing!

Further, you *will* have contention issues, Jet does not support record locking, it locks a chunk of the db file to imitate record locking; further still, Jet lacks a truly constructive way to deal with such contention. Left with a db engine paradigm that seems to be on the edge of workablility, you'll spend resources trying to work around it's shortcomings −− but it's all a waste.

Jet is not on the edge, it's over the unworkable line. About 8 years ago, I decided to never start anything new based on Jet, and to only work on Jet−based projects that are actively transitioning to SQL... quite possibly one of my best business/career decisions... but reaching that point cost me dearly.

> You have to realize this job is a poster child for "perfect world" vs "real world" scenarios.

Heh... db engine choices are about as "real world" as it gets! And with free alternatives to Jet, it has little to do with "perfect world" so that's an excuse, not a reason.

If the client approached you with Jet already chosen as the db engine, then the hard part is knowing enough to tell them, "no, it won't work reliably like this, bottom line, end of story." If that position loses the project, then you are better off without it.

> I have been using ADO for about 7 years
> now and have not had to face this issue before.

I think it likely that the problem is more with the Access links to a remote Jet database, than with ADO itself. If that is the case...

> Thank you all for your input. I will keep trying to find a way to handle this timeout.

Best of luck with that −− you're going to need it... It's unfortunate that each of us has to learn the hard way.

−Mark

> Thank you,
>
> Robert

> "Bob Barrows [MVP]" wrote:

I don't understand. You say it is "vital" that the information be passed to the central server, but you are willing to live with periods where it isn't sent?

I am tending to concur with Mark on this. You should have two processes, neither of which depends on the other:
1. The data collection process
2. the data aggregation process

In the data collection process, the data is collected and stored to a robust location, i.e. a local database or even a text file.

The data aggregation process passes the locally stored data to the central server. This process can consist of either a scheduled program that runs periodically (perhaps every 5 seconds?) or a program that is left running and polls the local store periodically for new data. You might even make use of MSMQ.

Robert McCarter wrote:

> Mark,
>
> Thank you for your reply. The only reason I am using Jet/Access is
> because it is the only means I have available at the current time.
> Trust me, I would much rather write stored procedures and use server
> side cursors. But I do not have a SQL server available at this time.
> I plan to propose adding a SQL server for the job as soon as I get
> the installation phase completed. However, that being said, I will
> still find myself up against corporate IT policies etc. I have full
> control over the Jet/Access infrastructure. If I have to play IT
> politics getting a SQL server installed, I will not even worry about
> trying to implement that solution. Like Sean Connery said in "The
> Untouchables" I won't be stupid enough to show up to a gunfight with
> a knife.
>
> At any rate thank you for your comments. I am trying to come up with
> the best solution for the technology I have in place. In regards to
> using the local database, it is vital that I get the information

to a
central server. There are 5 different machines running this
test
program that need to read the data from all tests. The only
reason I
have the local database in place is to allow the program to
continue
to function until the connection can be restored to the server.
Once
the connection is restored, the tables will be immediately
synchronized. In the interim, all sorts of bells and whistles
will
be going off to alert someone that there is a problem. −−
Thank you,

Robert

"Mark McGinty" wrote:

> "Robert McCarter"
> <RobertMcCarter@xxxxxxxxxxxxxxxxxxxxxxxxx>
> wrote in
> message
> news:FC2DADD5−B384−470D−931E−DBB91E8CCDA8@xxxxxxxxxxxxxxxx
>
> > Thanks for your reply Bob,
> >
> > I am using VB6 SP6 with a
> > reference to ADO 2.8. The
> > program that I
> > currently have in production
> > is checking for the
> > connection to the
> > database
> > outside of ADO land. I am
> > using some ICMP code
> > (found on
> > support.microsoft.com site)
> > to ping the server to ensure
> > that the
> > connection
> > to the server is ok. I am also
> > doing a DIR() statement to
> > verify
> > that my database exists on
> > the server. The problem I
> > am trying to
> > address is a performance

> issue. During a test, I might read or
> write to the database 10−15 times. Each time I attempt to read or
> write to the database, I do the
> following:
>
> 1. Verify Link To Database Is OK with ICMP Ping Test
> 2. Verify Database Exists On The Server
> 3. Open Connection To The Database
> 4. Perform Sql Execute or Recordset Read That Is Required
> 5. Close My Connection To The Database
>
> I coded my program in this manner such that when I do have a
> connection issue, I would not crash my program or even worse halt
> production. The particular application is running on a line that
> loses $38 per second of downtime. So if my program slows
> production by 10 seconds, I have cost the
> customer $380.

Forgive my candor, but if that's the case, why in hell are you using
Jet/Access as the db engine?

In any case, this is obviously not an OLTP application, if a local
db is acceptable for interim periods, so you might want to re−think
your architecture −− like maybe always writing the data to a local
db regardless, and using a second process to transfer the data in
chunks when the connection to the central

data store [I can't, in
good conscience, to refer to Jet as a "server"]
is available.

That would give you the fastest and most
reliable connection
possible by far, [and would be most free
from contention issues as
well] to service your real−time data
acquisition needs. The central
data store will be as current as the interval
for the background
process (plus any connection downtime, of
course), and the
background process will be in a position to
take its time with the
transfers, without costing anyone any
time/money.

That you're ever able to consider use of a
local database
necessarily means that getting the data to the
central store is not
important to the data acquisition process, so
off−loading that
responsibility to another process should give
you the best of all
worlds... Of course, whether or not that's
entirely practical, only
you can say...

But whatever you do, my advise would be to
seriously consider the
wisdom of basing the integrity of an
important activity, on
something that's widely known to be flakey
and replete with
inadequacies, as is Jet. IMHO, using Jet as
the cornerstone for
something that could cost someone
$38/second *when* [that's right,
not "if" but "when"] it nuts−up due to its
many inherent flaws, is
effectively *begging* for grief.

−Mark

What I am ultimately
attempting to accomplish is
this:
When the program loads, I
want to verify I can connect
to the
database. I then want to
establish a connection and
leave it open.
But, I need to monitor this
connection to check whether
my database
link is severed. This
is why I would like to be
able to programmatically
capture any
connection errors and
handle them accordingly.
This whole issue is
really a big WHAT IF
(which is the majority of
what you have to write your
code around),
but I would like to be able to
control any problems that
may arise.

The problem that I have
with the way I am doing it is
that Opening
and Closing a connection to
a database takes time.
Granted in most
applications
1−2 seconds is not a lot of
time, but in this application
it is very
critical.

Thank you for any insight
you may be able to offer.
−−
Thank you,

Robert

"Bob Barrows [MVP]"
wrote:

Robert McCarter wrote:

> Hello,
>
> I am trying to configure ADO to give me a trappable error when the connection to my Access database becomes unavailable. I created the following variables in a module and made them public:

Is this a VB application? Or in an Access VBA module?

> Public

MSCn
as
New
ADODB.Connection
Public
MSCd
as
New
ADODB.Command
Public
MSRs
as
New
ADODB.Recordset

In
my
startup
form,
I
set
all
of
these
variables
and
connect
to
the
database
etc.
What
I
am
trying
to
accomplish
is
this:
When
I
have
a
good
connection
the
remote
machine
that
holds
the

Access database (normal operation) everything runs normally. However, if the link to the remote database is severed (cable unplugged, database damaged, or some other problem) I want to use a database local to the machine running the program.

I have tried to set up a test program that connects to

the
remote
database
and
then
physically
unplugging
the
cable.
I
cannot
get
around
the
default
30
second
timeout.
I
would
like
a
timeout
of
5
seconds
to
try
the
main
server
and
if
I
get
an
error,
swap
over
to
the
local
database.
When
I
am
running
my
program,
and
I

unplug
the
cable
it
still
takes
30
seconds
to
get
an
error.
I
have
tried
ConnectionTimeout
on
the
connection
and
CommandTimeout
on
the
Command
and
had
no
luck.

Furthermore,
when
I
do
get
an
error,
it
is
one
of
those
messagebox
(Debug,
End)
errors.
I
would
like
to
make
a

global
error
trap
instead
of
having
to
code
"on
error..."
stuff
in
each
sub
since
I
use
the
aforementioned
variables
all
over
my
program.
I
tried
to
use
WithEvents
when
dimensioning
my
connection
and
could
not
get
the
InfoMessage
Event
to
fire.

Again ...
VB or
VBA?

I

would
appreciate
any
ideas
any
of
you
could
give
me.
This
program
is
to
be
run
in
a
production
environment
on
an
assembly
line
that
indexes
at
20
seconds
per
cycle.
So,
I
do
not
have
much
time
to
wait
for
timeouts
etc.
if
I
have
problems.

There is no
avoiding

this delay if
you use a
Connection
to test the
database
existence.
However, if
you use a
FileSystem
object, you
should have
better
success. Set
a Reference
to the
Windows
Scripting
Host and try
this:

Function
DBExists(byval
FilePath as
String) As
Boolean
Dim fso As
New
FileSystemObject
DBExists =
fso.FileExists(FilePath)
End
Function

−−
Microsoft
MVP −
ASP/ASP.NET
Please reply
to the
newsgroup.
This email
account is
my spam
trap
so I don't
check it
very often.
If you must
reply
off−line,
then

remove the
"NO
SPAM"

−−
Microsoft MVP −− ASP/ASP.NET
Please reply to the newsgroup. The email account listed in my From header is my spam trap, so I don't check it very often. You will get a quicker response by posting to the newsgroup.

.