

SSIS Scheduling Failure – AcquireConnection Error

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.dts/2006-05/msg00020.html>

- *From:* JJ of Eugene OR <JJwithQuestions@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 2 May 2006 10:06:01 -0700
-

In honor of the help that I have recently received in this group, I thought I would post the answer to a problem that had been plaguing me for a couple of weeks in the hopes that this posting will help someone else.

Like many people, I am in the middle of upgrading from SQL Server 2000 to 2005. The biggest hurdle is converting the DTS packages to SSIS. (Aside: I truly like many things about SSIS. I don't mind a relatively big learning curve if the end result is an easier to understand working environment as well as one that will let me do more things faster in the long run. On the other hand, it's definitely working like a first release for me. I'm looking forward to future releases, including true support for ODBC, conveniences, etc.)

Anyway, I have a story similar to a lot of others out there, but I couldn't find the answer to this particular problem on the internet. I had a package that ran fine in Business Studio. It also ran fine when I imported the SSIS package into SSIS services in Management Studio and ran the job manually. However, I got the dreaded 'AcquireConnection' error when I tried to schedule the job:

Error: 'The AcquireConnection method call to the connection manager 'conn_ORaccPublic_OLEDB' failed with error code 0xC0202009. '

I was very familiar in DTS with jobs running manually, but failing when scheduled. However, in DTS, the problem was always an issue of resources being different on my personal PC compared to the server. For example, if I ran a DTS package from my personal PC, DTS uses the ODBC drivers and ODBC DSNs on my personal PC to run the package. When I scheduled the package to run on the database server, however, SQL Server uses the resources on the server. If the server didn't also have the same ODBC driver and same ODBC DSNs, then the job would fail.

The difference in my SSIS packages is that I have been doing the development while logged onto the SQL Server 2005 database server itself. All the resources available when I ran the job manually should also have been available when I scheduled the job.

What was going on? After doing a bunch of tests and research, I finally

SSIS Scheduling Failure – AcquireConnection Error

discovered something of the meaning behind the 'run as' box in the 'general' property page for the job. It looks like I was experiencing a permissions issue or something like that. When I ran the job manually, I was running it as myself. But when I scheduled the job, the default value in 'run as' is the 'SQL Agent Service Account'. Can't say that I understand much of what that account is or does, but it lead me to the first 'fix' for my problem:

- a) add a new 'credential' that had my own account attached to it
- b) a new 'proxy' that used the newly created credential
- c) changed the value in the 'run as' box to the newly created proxy

I don't want my account to be the final answer for this problem, but it at least got the job running when scheduled – well sort of. After a little victory dance when the scheduled job ran a few minutes into the future, I scheduled it to run in the middle of the night like it will have to do when it goes into production. This was a mere formality, but I like to be thorough. And guess what?!? The job failed that night! And I got the same error.

** Running under me as TestProxy allowed me schedule the job that day, but when run at 3:15 AM that night, I got the same error again.

—Theory: The error message talks about having an 'AcquireConnection' problem and issues with the 'connection' manager. The SSIS test job was running at the same time as my production DTS job. Maybe the production job was running a second first, getting locks on the tables, and preventing the SSIS job from making that all important connection.

Test: run SSIS job at 2:14 AM, an hour earlier. Since the job takes less than 3 minutes to run, this should prevent all conflict between the DTS job and the SSIS job.

Result: still failing with same error message.

—Theory: The schedule in the job itself was corrupted somehow or a bug tied it directly to a bad version of the SSIS package. This made a little sense because of a web page I read about someone making a change to an SSIS package (or a maint plan?) or a schedule and the changes not being recognized in the other place. Or something like that. Another somewhat supporting reason for this theory: I had logging turned on for the package. In addition to the OnError event, I eventually had the package log the pre and post validation events. I could see that when the job was failing, it was validating tasks in a different order than when the job was succeeding, despite the job running the exact same package each time.

Test: manually create a new schedule in the job to run the package that night (still at 2:14 AM).

Result: still failing with same error message.

—Theory: Perhaps the current problem is related to security and proxies just like it was the first time the error appeared. When the job succeeded, it was a situation of me scheduling the job to start a few minutes in the future. During the time that the job actually ran, I was logged onto the database server or at least our network. Since the job is running under a proxy that is related to my windows login/account, maybe the job was

SSIS Scheduling Failure – AcquireConnection Error

succeeding during the day time, because I was logged onto the network—and failing at night because I was logged off of our network. (Hey, I'm getting desperate here.)

Test: Set job to run 15 minutes in the future. Then, right away, completely log off of all computers on the network and go to a meeting.

Result: When I logged back onto my computer an hour later, the job/package had run just fine. So, whether or not I am logged onto the network does not seem to explain why the job is running fine in the daytime and failing when I schedule it to run at night. [Don't stop here, because this didn't actually prove to be true.]

—Theory: Perhaps this issue is related to both timing and our network situation. I know that the source database is connectable during the times I have tested since the DTS package runs just great. But what if the timing issue is somehow related to my account being active/available at certain times or maybe something is being done to our networks or test database server at night that I don't know about. Can I narrow down the timing when the job works and when it doesn't?

Test: Schedule the job to run every 20 minutes from 4:50 PM in the evening to 7:30 AM the next morning. I logged off the network about 4:30 and got back on the next morning about 7:25.

Result: EVERY SINGLE run failed.

—Theory: I'm back to thinking that this has something to do with me being logged onto the computer or not. Perhaps the test I did before was not sufficient. I remembered that in the past, when administrators have changed some property of my account, there was sometimes a request that I wait 15 or so minutes for the property to work its way into the system (or something like that). So, what if I just hadn't waited long enough in my previous test to see if me logging off was the issue or not.

Test: Create two schedules to run the job. Each schedule ran the job every 20 minutes with these time brackets: from 3:10 PM to 9:00 PM and 6:10 AM to 9:00 AM the next morning. During the day, I am typically logged onto both my own PC and remotely logged onto our database server. I logged off both computers at 4:35 PM that afternoon. When I returned to work the next morning at 7:20 AM, I logged onto my own PC only. I waited until 8:13 (after the 8:10 run) to remotely log onto the database server.

Result: You can see the schedule below.

The timeline went like this:

3:00 – I am logged onto both my PC and the database server

3:10 – first job runs with success

3:30 thru 4:30 – the jobs continue to run with success

4:35 – I log off both computers

4:50 – the job runs with *success* (remember that the day before, when I had run the job at 4:50, but logged off at 4:30, the job had failed)

5:10 pm thru 8:50 pm – the job runs with failure

6:20 am thru 7:10 am – the job runs with failure

7:20 – I log onto my own PC, but not the database server

7:30 thru 8:10 – the job runs with failure

8:13 – I remote log onto the database server (though I did not open

Management Studio or connect to the database server thru Management Studio =

SSIS Scheduling Failure – AcquireConnection Error

one step at a time)

8:30 – job runs with success!!

CONCLUSION: It does indeed look like the problem is related to me being logged onto the database server or not. If I am logged onto the server itself or within 15 minutes of logging off, the job will run fine.

FYI: Issues I Am Still Working On:

- a) What is it about my account that makes the job run? In other words, what permissions or properties of my account do I need to replicate into another non–personal windows account that can be used to make the SSIS job run in the future?
- b) How do I deal with the issue of logging on in the middle of the night? Do we have to have this new 'proxy can run SSIS jobs' windows account logged into the database server at all times or is there a way to get the database server to treat this new proxy windows account the same way it treats the default 'SQL Agent Service Account' as always being 'on'?

Notes:

When the job fails, I can see that it fails at the same data flow task each time. This is true whether the failure occurs because I am running the job with the 'SQL Agent Service Account' or because I am running the job with my special TestProxy and yet am scheduling it for the middle of the night.

When the job fails, I get the following events in the log file:

- 1) prevalidate the package
- 2) OnPreValidate a data flow task (the same one that it sticks at every time when failure occurs)
- 3) OnError for the above data flow task. There are actually four OnError records for this task. The error messages are:
 - a) 'The AcquireConnection method call to the connection manager 'conn_ORaccPublic_OLEDB' failed with error code 0xC0202009. '
 - b) 'component 'OLE DB Source' (1) failed validation and returned error code 0xC020801C. '
 - c) 'One or more component failed validation. '
 - d) 'There were errors during task validation. '
- 4) OnPostValidate for the data flow task

(No post validate for the package appears in the log.)

The data flow task that is failing is neither the first task in terms of 'run' or 'path' order or in terms of alphabetical order. It is also not the first or last task to be validated when the package is run successfully. When the package is run successfully, the validation for the data flow task in question happens in the middle of the log rows.

--

– JJ, Eugene OR

.