

Re: Driver AutoCommit issue

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.jdbcdriver/2006-05/msg00023.html>

- *From:* Joe Weinstein <joeNOSPAM@xxxxxxx>
 - *Date:* Thu, 04 May 2006 11:22:10 -0700
-

Arun wrote:

Joe

I am not calling the connection.commit(). I am using the container managed transaction and I expect the container to handle that for me. Is there different driver class that i have to use in the connection pool configuration.

Well, I would suggest using the com.microsoft.sqlserver.jdbc.SQLServerXADataSource, but the main issue is that you should never call setAutoCommit() or any other transactional stuff if you want the container to be doing that stuff, and crucially, your DataSource for accessing the pool has to be a TxDataSource so we know to include your JDBC in the tx if any.

However, even so, you should alter your code like mine, to try doing a rollback immediately after calling the procedure, and then do a query to see if the change happened or not. And I do suspect that your DAO code may not be thread-safe for use in WebLogic.

Joe

"Joe Weinstein" wrote:

Arun wrote:

Joe,
Thank you for you quick reply

Why I am saying the DML always gets committed is when I step thought the code I can see the updated data in the

Re: Driver AutoCommit issue

"Joe Weinstein" wrote:

Arun wrote:

I use MS SQL Server latest JDBC driver and trying to connect to MS SQL Server 2000 SP3 database using the datasource. I set the auto commit to false on connection as soon as I get the connection from the datasource. For some reasons what ever DML statements I execute through the connection it gets committed to the database. below is the snippet of the code that I am trying to execute.

```
connection =  
mydataSource.getConnection();  
connection.setAutoCommit(false);  
logger.debug("Connection  
Auto Commit Status "+  
connection.getAutoCommit());  
CallableStatement  
callableStatement =  
connection.prepareCall(  
{call dbo.updateData}");  
callableStatement.execute();
```

The logger printed :
Connection Auto Commit
Status false.
dbo.updateData is a very
simple procedure that
updates a table. There not
begin transaction or commit
transaction in stored
procedure.

Re: Driver AutoCommit issue

How are you proving the data is committed?
I just ran this code and
got what you'd expect:

```
DatabaseMetaData dd = c.getMetaData();
System.out.println("Driver version is " +
dd.getDriverVersion() );

Statement s = c.createStatement();
try{s.execute("drop table foo");} catch
(Exception ignore){}
try{s.execute("drop procedure bar");} catch
(Exception ignore){}
s.execute("create table foo(qwe int)");
s.execute("create procedure bar as insert foo
values(1)");

PreparedStatement p =
c.prepareStatement("{ call bar()}");

c.setAutoCommit(false);
p.executeUpdate();
c.rollback();

ResultSet r = s.executeQuery("select
count(*) from foo");
r.next();
System.out.println("We inserted " +
r.getInt(1) );
```

I got:
C:\new_ms_driver>java foo
Driver version is 1.0.809.102
We inserted 0