

Re: Driver AutoCommit issue

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.jdbcdriver/2006-05/msg00024.html>

- *From:* Arun <Arun@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 4 May 2006 12:29:03 -0700
-

Joe

Isn't the XA driver for the distributed transaction, I need to install the path sqljdbc_xa.dll and xa_install.sql before using the XA driver. i.e. Since I will be accessing a production system I will not be able to install additional DLLs.

I tried without installing the patch got some missing stored proc error when the connection pool was getting created.

I created a TX aware datasource using com.microsoft.sqlserver.jdbc.SQLServerDriver and tried to get the connection from the TX aware data source. But this time I got a different exception. This time the exception is because I tried to use two connection pools in my application one to connect to MS Sql server and another to connect to Oracle. If I don't use a XA driver I get "Connection has already been created in this tx context for pool" exception and if I use the XA driver for SQL Server without installing the xa_install.sql I get missing stored proc error. Is there a way to use the non XA SQL server ConnectionPool along with the Oracle Non XA Connection pool.

"Joe Weinstein" wrote:

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,

Re: Driver AutoCommit issue

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 your quick reply

Why I am saying the DML always gets committed is when I step through the code I can see the updated data in the database immediately after the callable statement is executed. Even before the EJB method that invoked the call is completed.

I am not using the driver from a stand alone application, I am using the driver from within weblogic. I have configured the datasource in weblogic config.xml using the following configuration

```
<JDBCConnectionPool
DriverName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
Name="myPool"
PasswordEncrypted="{3DES}ifMhpj12kfwKJ+1BteiLZg=="
Properties="instanceName=SQL Server
2000;user=nemoadap;url=jdbc:sqlserver://my_db_host_name:1433;userName=user;da
Targets="admin" TestTableName="SQL
SELECT 1"
URL="jdbc:sqlserver://my_db_host_name:1433"/>

<JDBCDataSource
JNDIName="myDatasource"
```


Re: Driver AutoCommit issue

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
```

Re: Driver AutoCommit issue

```
Commit
Status "+
connection.setAutoCommit());
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.

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
```

Re: Driver AutoCommit issue

```
(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
```