

## FirebirdSQL Fixes Bug in JDBC Driver For getupdatecount()

Posted At : August 25, 2003 6:52 AM | Posted By : Steven Erat

Related Categories: Java, Quality Assurance, ColdFusion, Macromedia

Users of the open source database project **FirebirdSQL** should take note that the Firebird developers have just fixed an important bug in the **Jaybird JCA JDBC** driver.

It had been reported on the **forums** and in mailing lists that when using ColdFusion MX and MX 6.1 that INSERT and UPDATE sql statements in ColdFusion queries would hang and spike the cpu, even though the sql did in fact complete such that the new record was inserted or an existing record was updated.

One user on the forums reported today:

Good news! We reported this problem to the FireBird developers last week. The updated driver and class files have been released and we tested this morning. So far, we have been able to select, update and insert with the new driver.

Stephen Dupre of Macromedia QA reported a brief explanation and stack trace sample:

The insert works OK but the driver is never returning -1 for `getUpdateCount()` - so it loops forever. I even see `jrun.exe` not quite spinning but taking up ~30% of the CPU on my Dual Pentium system.

The stack trace shows:

```
coldfusion.runtime.RequestTimedOutException: The request has exceeded the allowable time
limit Tag: CFQUERY
at coldfusion.sql.Executive.getRowSet(Executive.java:486)
at coldfusion.sql.Executive.executeQuery(Executive.java:974)
at coldfusion.sql.Executive.executeQuery(Executive.java:886)
at coldfusion.sql.SqlImpl.execute(SqlImpl.java:229)
at coldfusion.tagext.sql.QueryTag.doEndTag(QueryTag.java:447)
..
```

The call is spinning at `getUpdateCount()`. We've seen other drivers that don't follow the spec with similar behavior.

In this case, the driver was not following the JDBC spec where it should return a value of -1 on `getUpdateCount()` when all inserts or updates were complete.