# EML - Bug #2576

## Data Manager Library: Database Connection Pooling

10/27/2006 01:11 PM - Duane Costa

| | | | |
|---|---|---|---|
| **Status:** | In Progress | **Start date:** | 10/27/2006 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Jing Tao | **% Done:** | 0% |
| **Category:** | datamanager | **Estimated time:** | 0.00 hour |
| **Target version:** | DataManager 1.0.0 | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 2576 | | |

**Description**

Rework the design and implementation of database connection pooling in the Data Manager Library. Provide a callback mechanism for the calling application to manage its own connection pool. This should include a mechanism for returning a "Connection not available" status to the Data Manager so that it will know that it needs to wait until a connection is available. The Data Manager should generally use one connection per operation, though if the operation has several steps it could re-use the same connection in more than one step if it's safe to do so.

### History

#### #1 - 10/09/2007 03:47 PM - ben leinfelder

As part of the work done on bug #2979, it came to light that there was a significant bottleneck when hitting the database to look up entity/attribute names to generate SQL queries.
An alternative to the org.ecoinformatics.datamanager.database.DatabaseConnectionPoolInterfaceTest implementation of DatabaseConnectionPoolInterface was created: the org.ecoinformatics.datamanager.database.pooling.* classes along with [another] properties file (pooling.properties). The DatabaseConnectionPoolInterface implementations (currently HSQL and Postgres) use the connection pooling provided in their respective libraries and *dramatically* enhanced performance (~tenfold reduction). A calling app would rely on the DatabaseConnectionPoolInterfaceFactory (and correct settings in pool.properties) to provide a DatabaseConnectionPoolInterface instance.
Feedback on this approach is appreciated as I'm not certain it meets all the requirements originally specified in this bug.

#### #2 - 01/11/2010 11:19 AM - ben leinfelder

We might also want to exploit third party connection pooling resources - web app containers and application context providers come to mind, maybe the apache commons pooling tool.
At one point I experimented with hooking into Metacat's existing (roll your own) connection pooling, but it proved more difficult than I'd anticipated and also blurred the lines between the different databases in a way that seemed slightly risky given the newness of DML.
As the DML is exercised, I think it'll be more obvious if and how additional connection pooling strategies can be used.

#### #3 - 03/27/2013 02:20 PM - Redmine Admin

Original Bugzilla ID was 2576