

EML - Bug #2575

Data Manager Library: Support for query object API

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

Status:	Resolved	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:	2575		
Description			
<p>The original design of the Data Manager Library was somewhat vague in its support for querying data tables. After further discussion (Matt, Jing, Duane, and Mark Servilla), we think that allowing the calling application to pass in an ANSI SQL string would be too problematic because of the parsing requirements. The problems arise from needing to parse non-standard entity names into database table names, and non-standard attribute names into database field names. For example:</p> <p>SELECT SPECIES NAME, SPECIES ID FROM SPECIES</p> <p>means one thing from the perspective of entities and attributes, but something else from a database perspective, where "NAME" and "ID" would be interpreted as column aliases.</p> <p>Instead, we will design a query class that the calling application can use to construct its queries in a more structured way by setting various attributes of the query object. At some later point, we may also support queries in an XML format that could be mapped onto the query object by the Data Manager Library. This would facilitate passing queries between two or more processes (e.g. first from Morpho to Metacat, and then from Metacat to the Data Manager Library code).</p> <p>The JDBC ResultSet object that is returned could also pose a problem, since it contains references to the database table field names, not the original attribute names. The calling application could get around this by restricting itself to accessing the fields by position rather than by name.</p>			

History

#1 - 01/11/2010 11:38 AM - ben leinfelder

There are Java and XML query specifications at this point.
The Java API abstracts the SQL-specifics quite well (save for some reserved DB words as mentioned in bug [#2737](#)).
The XML goes one step further and allows you to pass an XML representation of the query to the DML where it is parsed and turned into the Java version of the query.
Both mechanisms return a ResultSet to the calling application. There are ways to get at the resultset columns using attribute name/label/index translations and this can be useful when dealing with synthetic datasets (where many tables have been joined). I agree that the "safest" way to access the columns may be by index because of the fact that EML allows us to reuse attribute names.

Unless I've misread this bug, I believe it is resolved. Please reopen if not.

#2 - 03/27/2013 02:20 PM - Redmine Admin

Original Bugzilla ID was 2575