Subject: Re: Firebird support
Posted by AJM on Fri, 27 Apr 2007 14:06:06 GMT
View Forum Message <> Reply to Message

Here are my replies to your questions.

1. Access to a single object per DBMS is controlled in the _getDBMSengine() method of the abstract table class (std.table.class.inc). That method includes the database name as an argument, so it may be possible to amend this so that for firebird it maintains a separate instance for each database instead of one instance for all possible databases. However, this does not get round the fact that in some places I actually use sql queries which contain JOINs to other databases. If Firebird cannot handle this then it probably makes it unusable as a DBMS for Radicore.

2. The record count for each sql SELECT is absolutely essential. This is divided by the page size in order to get the page count. If Firebird cannot obtain this count efficiently then it is a black mark against Firebird.

3. All my drivers contain a *_free_result() call. It is not absolutely necessary as every resource is automatically freed when the script ends.

4. Radicore does not ensure that you only have one autoincement field per table. If it is valid for the DBMS then it is valid for Radicore.

5. Radicore does not use exceptions. The existing error handling works without exceptions, and I see no advantage in changing it.

6. Radicore does not “require” any locking. It is up to each DBMS driver to implement whatever locking choices are available via the _setDatabaseLock() method.

7. Take a look at the startTransaction() method inside std.table.class.inc which subsequently calls the startTransaction() method in the DBMS driver. The first it database agnostic while the second is database specific.

8. No comment.

9. Why don’t you see how any of the existing database drivers handle it? As far as I can see if it is a valid sql statement then Radicore will deal with it.

10. Yes. Every primary key field is a required field, so unless it is an auto_increment field its absence will be detected in the _validateInsert() method.

11. No comment.

12. No comment.

When it comes to porting the schemas and the data there is no quick way. You will have to copy and manually alter the existing sql scripts.