Subject: Re: Firebird support
Posted by ptimmermans on Tue, 01 May 2007 10:03:12 GMT
View Forum Message <> Reply to Message

Hi,

First of all, the 'pecularities' I mentioned in my earlier post were not meant to offend you or anyone, but simply some of the findings I encountered when porting Radicore to Firebird. Even more so, I like the approach you took to create a dictionary instead of generating/interpreting code run-time like RoR is doing based on conventions. Furthermore, keeping the tiers separate is IMHO a big plus, especially with a workflow system on top. But I first have to evaluate it to see whether it fits my situation. That is why I am now first trying to port Radicore to Firebird.

The answers you gave, though, might throw some sand in the machine:
1. Firebird does not allow multi table queries in multiple databases, at least not by joining them! From the code I saw that the menu tables are queried from the dict database. Are there other situations? Are all the databases (dict, menu, workflow and audit) actually needed run-time? Is it possible to store the tables in one Firebird database together with the application tables? This would also solve the switching to different databases problem I mentioned.
2. Usually, computing the COUNT in Firebird is fast and simple, but in complex queries it could take time. So I guess it will not be a real problem except in certain cases. One simply has to be aware of that.
3. There are indeed _free_result/_free_statement calls but not all over the code. E.g. in the deleteRecord in dml.mysql.class.inc there is no mysql_free_result after querying the mysql_affected_rows. I was wondering if this is on purpose or simply because (BTW I like closures, create-destroy, open-close, etc.).
4. In the dml.mysql.class.inc there is a local variable $auto_increment that saves the autoinc primary keyfield. But there is only room for one field. So that is why I wondered if/how Radicore checks for at most one autoinc field? Even more so, if there are more autoinc fields, then only the last will be recorded in the local variable.
5. Concerning exceptions: in menu\logon.php a startTransaction is performed, followed by a user_logon. The user_logon might fail during a getData_raw (triggering a _dml_getData), causing the script to halt. But what about the transaction? Is it rolled back/committed? It depends on the database and the default transaction behaviour in case a PHP object goes out of scope (and is being destroyed). I personally like explicit programming and not depend too much on implicit behaviour. By wrapping the menu\logon.php in a try-catch you can release resources explicitly. This might be valid for other situations in your code as well. Exceptions are new to PHP 5, as I understand. So that is why I wondered if/when you are going to use them. I think they are worth investigating to make code more robust.
6. I am glad that no locking is required.
7. OK. Transaction levels are an Oracle thing only.
9. What I mean is that the getCount has a $where parameter that is either a SELECT or a WHERE, but nowhere is the $where checked (in case of a SELECT) that only one aggregated column value is returned. So, it is up to the developer to verify that a given SELECT statement is valid for the getCount.

Still busy, converting MENU, WORKFLOW and AUDIT.

BTW, I like the amount of documentation on the Radicore website regarding. Also the discussions concerning a.o. object oriented programming. My personal view is that theory and practice should be synergetic. One should not rank one above the other. What good is theory without proper application? And building an application that works does not make it a 'good' application perse (it needs to be maintainable, robust, performing, functional etcetera). One needs vision (where having a theorical background can help) but also hands-on mentality and perseverence, and the discipline of constantly looking in the mirror to see whether things can be made simplier (thereby using theory and experience). Learning by example from people that have walked a proven path before is also a valid approach. That is why I am currently looking at Radicore. What are the good things, and where could it become even better.

But I am looking at other sources/documents too. Did you ever take a look at http://msdn2.microsoft.com/en-us/library/ms978496.aspx regarding multi-tier development?
