Subject: Firebird support
Posted by ptimmermans on Wed, 25 Apr 2007 07:02:39 GMT
View Forum Message <> Reply to Message

Hi,

I have downloaded the Radicore package and I am now in the process of getting Radicore to support the Firebird open source RDBMS. I was wondering what steps I have to take (and in what order) to port Radicore to support Firebird?

I started with the Data Dictionary. I have created a folder radicore\dict\sql\firebird and reworked the MySQL/Oracle files dict-schema.sql and dict-data.sql. Then I created the Data Dictionary database DICT.FDB (I attached a zipfile of these files to this message).

What I understood from the documentation is that I have to create the Audit, Menu (RBAC), and Workflow databases as well? I assume that it is possible to generate the ???-schema.sql from the Data Dictionary? But how? I probably first have to port the includes\dml.<database>.class.inc file, so I started with a dml.firebird.class.inc file, based upon dml.mysql.class.inc. But Firebird is not compatible with either MySQL or Oracle. E.g. there is no ibase_num_rows, no autoinc fields (but sequences like Oracle). So porting is not that straightforward.

Please, I need some help.

File Attachments
1) dict.zip, downloaded 2018 times

Subject: Re: Firebird support
Posted by AJM on Wed, 25 Apr 2007 08:57:45 GMT
View Forum Message <> Reply to Message

The first thing to need to do is to create a dml.firebird.class.inc file so that you can call the APIs which communicate with that DBMS. Leave interrogation of the INFORMATION_SCHEMA (used used by the data dictionary 'import' functions) till later.

The second step is to take the existing schemas for the MENU, AUDIT, WORKFLOW and DICTIONARY databases, modify them as necessary, then load them into firebird. You may also want to do this with the prototype subsystems as well.

The third step is to load all the data (from the xxx-data.sql files) into the relevant databases.

The fourth step is to change your CONFIG.INI file so that GLOBALS['dbms'] has the value 'firebird'. It is IMPORTANT that this value matces EXACTLY what appears between 'dml.' and '.classes' in the file you created with the first step.

Having done that you should be able to logon to radicore so that you can check that it handles all SELECTs, INSERTs, UPDATEs and DELETEs correctly. It does not matter that at this point you will be using <table>.dict.inc files which were created from a different DBMS.

When this has been done the last step is to modify the various ddl_???() methods in your new database driver so that the dictionary import functions will work.

If you have any more questions please let me know.

Subject: Re: Firebird support
Posted by ptimmermans on Fri, 27 Apr 2007 10:29:53 GMT
View Forum Message <> Reply to Message

Hi,

Well, I created the dml.firebird.class.inc file (see attachment) but did not yet implement the _ddl functions. While implementing I came across the following pecularities:

1. Switching databases in Firebird means dropping the current database connection and connect to the next database. However, it is possible to have multiple (simultaneous) database connections open. In Radicore the Data Access Object can only have one connection. What if you need to access the Data Dictionary, Menu, Workflow, Audit and application database simultaneously?
2. Firebird has a so called multiversion architecture, and hence, cannot determine the number of records after issueing a SELECT, other than by walking the result set and counting the number of records (which is an expensive operation). The only other option is to issue a second SELECT statement, namely a SELECT COUNT(*), with the same FROM and WHERE clause. Unfortunately, this statement is also expensive because the result set still needs to be walked through. If the WHERE references fields that are indexed then the SELECT COUNT(*) can be fast. Furthermore, to have a reliable count the SELECT statements must both be executed within the same transaction. If this is not required they can be executed in different transactions, one after the other (Radicore works with pages of records and not the exact recordcount).
3. I added ibase_free_result() calls to release any database objects. Is this not required for MySQL or Oracle?
4. How does Radicore ensure that there can be only one AUTOINC field per table? Firebird does not support AUTOINC fields. Instead Firebird uses generator/sequence values similar to Oracle.
5. The Data Access Objectlayer does not (yet) use exceptions. Will this change in the future?
6. Firebird has a SELECT..FOR UPDATE but does not recommend using it for several reasons. So, no record/table locking is implemented. But, does Radicore require it?
7. What does Radicore do with transaction levels?
8. I still have to prevent SQL injection.
9. What if the SELECT statement in the getCount() refers to a non-aggregated column? Usually, it will refer to a COUNT(..), MAX(..), etc.
10. Does Radicore ensure that the primarykey fields are a subset of the fieldarray in insertRecord()?
11. Still need to implement BLOB functionality in insertRecord/updateRecord.
12. Still need to implement the _ddl functions.

Could you comment on these, please.

Please take a look at the attached file as well, and maybe give some hints how to improve it.

Next, I will be porting the MENU, WORKFLOW and AUDIT schemas and data. Is there a quick way of doing this?

File Attachments

1) dml.firebird.class.inc, downloaded 1849 times

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 â€œrequireâ€• 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 donâ€™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.

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?

Subject: Re: Firebird support
Posted by AJM on Wed, 02 May 2007 10:04:29 GMT
View Forum Message <> Reply to Message

1. There are numerous places where tables from different databases (or â€œschemasâ€• in PostgreSQL and Oracle) are accessed at the same time, sometimes within a JOIN but mostly through their separate classes. The MENU database is accessed within every script in order to verify the userâ€™s access to the selected task, and to extract the information necessary to build the menu bar and the navigation bar. The AUDIT database is accessed during every database update to record was has changed. It would be possible to move all the tables into a single database, but it would take a tremendous amount of work. All the while I have been building administrative applications, which is over 20 years now, it has been standard practice to give each subsystem its own database and its own code area. This can be handled with MySQL, PostgreSQL and Oracle, but if Firebird cannot support separate databases or schemas then it sounds like it is not suitable for use with Radicore.
2. Radicore allows you to construct whatever sql queries you like, and the performance of those queries is outside Radicoreâ€™s control.
3. It is not necessary to use *_free_result or *_free_statement all the time as all resources are automatically released when the script ends. Where the resource is potentially large, e.g. after a getData(), it is released as soon as it has been processed, but in other cases I do not bother. This

does not cause a problem, so I have no plans to change my code.
4. It is standard practice to have no more than one autoincrement column in a table, and to use that within the primary key. I have constructed the MySQL, PostgreSQL and Oracle drivers accordingly. Anybody who uses more than one autoincrement column needs to rethink their database design.
5. All fatal errors are routed through the error handler which is contained with file error.inc, and if you look you will see that if a transaction is in progress it will automatically perform a rollback. I have no plans to change Radicore to use exceptions for the simple reason there would be no benefit. The existing system works well, and changing it would not make it work any better. The Radicore code base works with PHP 4 which does not have exceptions, so that rules it out completely.
6. Database locking is described in http://www.tonymarston.net/php-mysql/infrastructure-faq.html #faq48
7. Different databases may have different or variable transaction levels, and how these are implemented is the responsibility of the individual database driver.
8. ---
9. It is not up to the framework to validate that an SQL query that you have generated is either valid or efficient. That is entirely up the developer.

I have never read any Microsoft articles on multi-tier development as they are oriented around Microsoft products which I do not use. I first read about the three-tier architecture when I was working with an obscure language called Uniface. The team I was working with designed a framework that was so horrendous the client cancelled the contract. I built my own version in two weeks (by modifying my existing two-tier framework) which outperformed what had taken them three man-years. The problem was that their framework was based on theory while mine was based on practical experience. Ever since that moment I have stopped listening to these â€œexpertsâ€• with their wild theories and done things my own way. The result is Radicore.

Subject: Re: Firebird support
Posted by zamolxes on Wed, 25 Jul 2007 20:57:55 GMT
View Forum Message <> Reply to Message

I wonder if ptimmermans was succesful in using Firebird with he RADICORE framework, since I am planning do do exactly the same.

Thanks for your feedback.

Regards from Transilvania

Horst

Subject: Re: Firebird support
Posted by AJM on Wed, 25 Jul 2007 22:28:22 GMT

View Forum Message <> Reply to Message

I believe that a Radicore driver for Firebird is a non-starter for the simple reason that Firebird does not allow access to more than one database schema per connection, unlike more mature databases such as MySQL, PostgreSQL and Oracle. Radicore uses separate schemas for each subsystem, and has many places where a single SELECT statement contains JOINs to several different schemas at the same time.

Before you ask the question the answer is no. I am not going to change Radicore so that it puts all its database tables into a single schema.

Subject: Re: Firebird support
Posted by zamolxes on Wed, 25 Jul 2007 23:05:20 GMT
View Forum Message <> Reply to Message

Hello Tony,

thanks a lot for your quick reply.

No, I was not going to ask you to change your concept, but instead the Firebird people to improve theirs.

I have been in the IT business (application and database design and implementation) for 38 years now, am familiar with the works of Edgar Codd and Chris Date, and I understand your design objectives quite well.

I found Radicore by chance these days, and I wish I had come across it earlier. I have wasted a considerable amount of time on RoR, only to find that RoR is not what I was looking for.

I have read several of your postings on the subject and I share your opinions on RoR.

As for Firebird, I do not depend on it, but can use MySQL instead. It would just have come in handy if it had worked.

Best regards

Horst

Subject: Re: Firebird support
Posted by AJM on Thu, 26 Jul 2007 09:05:44 GMT
View Forum Message <> Reply to Message

Are you aware that Radicore supports PostgreSQL and Oracle as well, so you are not restricted to MySQL only?

I would love to hear your comments on RoR, especially on those areas where RoR sucks and Radicore rules. You can post them to this thread Is Radicore better than Ruby on Rails?

Subject: Re: Firebird support
Posted by zamolxes on Thu, 26 Jul 2007 09:42:57 GMT
View Forum Message <> Reply to Message

Yes, I'm aware that it supports PostgreSQL and Oracle as well. I had been working with Oracle from Version 4 to Version 8. I will use MySQL, because it's the only of the three that is supported by my ISP.

I will gladly write about Radicore vs. RoR, as soon as I get some experience with it. I've only downloaded it yesterday, and will have to install and use it. So it may take a little while before I can make comments based on actual experience.
