
Subject: Firebird

Posted by [Novo](#) on Thu, 30 Jun 2011 04:16:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

I made an initial version of a firebird driver for UPP. Below is what it can do at this time. This code was tested on Windows with local connection to firebird 2.1 so far.

```
#include "firebird/firebird.h"

using namespace Upp;

void TestInt(FBSession& s)
{
    Sql stmt(s);
    stmt.SetStatement(
        "DROP TABLE test_long_table"
    );
    stmt.Run();

    // Data definition statement.
    stmt.SetStatement(
        " CREATE TABLE test_long_table( \n"
        "  id integer, \n"
        "  first_field integer DEFAULT NULL, \n"
        "  second_field integer DEFAULT NULL \n"
        ") "
    );
    stmt.Run();

    stmt.SetStatement(
        " INSERT INTO test_long_table \n"
        "  VALUES(?, ?, ?)"
    );

    for (int i = 0; i < 100; ++i)
    {
        stmt.SetParam(0, i);
        stmt.SetParam(1, i * 10);
        stmt.SetParam(2, i * 100);
        stmt.RunX();
    }

    stmt.SetStatement(
        " SET TRANSACTION WAIT ISOLATION LEVEL READ COMMITTED"
    );
    stmt.Run();
}
```

```

int _0 = -1;
int _1 = -1;
int _2 = -1;

Ref r0(_0);
Ref r1(_1);
Ref r2(_2);
int i;

// SELECT without parameters.
stmt.SetStatement(
    " SELECT * FROM test_long_table ORDER BY id"
);

stmt.Run();
i = 0;
while (stmt.Fetch())
{
    stmt.GetColumn(0, r0);
    stmt.GetColumn(1, r1);
    stmt.GetColumn(2, r2);

    ASSERT(_0 == i);
    ASSERT(_1 == i * 10);
    ASSERT(_2 == i * 100);
    ++i;
}
ASSERT(i == 100);
ASSERT(!stmt.Fetch());

// Rerun SELECT.
stmt.Run();
i = 0;
while (stmt.Fetch())
{
    ASSERT(stmt[0] == i);
    ASSERT(stmt[1] == i * 10);
    ASSERT(stmt[2] == i * 100);
    ++i;
}
ASSERT(i == 100);
ASSERT(!stmt.Fetch());

// SELECT with parameters.
stmt.SetStatement(
    " SELECT * FROM test_long_table WHERE id < ? ORDER BY id"
);

```

```

stmt.SetParam(0, 50);

stmt.Run();
i = 0;
while (stmt.Fetch())
{
    stmt.GetColumn(0, r0);
    stmt.GetColumn(1, r1);
    stmt.GetColumn(2, r2);

    ASSERT(_0 == i);
    ASSERT(_1 == i * 10);
    ASSERT(_2 == i * 100);
    ++i;
}
ASSERT(i == 50);
ASSERT(!stmt.Fetch());

// Rerun SELECT with parameters.
stmt.SetParam(0, 25);

stmt.Run();
i = 0;
while (stmt.Fetch())
{
    stmt.GetColumn(0, r0);
    stmt.GetColumn(1, r1);
    stmt.GetColumn(2, r2);

    ASSERT(_0 == i);
    ASSERT(_1 == i * 10);
    ASSERT(_2 == i * 100);
    ++i;
}
ASSERT(i == 25);
ASSERT(!stmt.Fetch());
}

CONSOLE_APP_MAIN
{
    const Vector<String>& cmd_line = CommandLine();
    if (cmd_line.GetCount() > 0 && FileExists(cmd_line[0]))
    {
        FBSession s;
        s.Connect(
            cmd_line[0],
            NULL,
            "SYSDBA",

```

```
"masterkey"  
);  
  
TestInt(s);  
TRANSACTION(s) {  
    TestInt(s);  
}  
}  
}
```

Actually, it can do more than that. It should support all data types except of blobs and arrays.

It also introduces an interesting macro called TRANSACTION. An example:

```
FBSession s;  
  
TRANSACTION(s) {  
    TestInt(s);  
}
```

If code inside of a TRANSACTION block successfully reaches end of the block, then transaction will be committed. If for some reason code jumps out of the block (in case of an exception, for example), then transaction will be rolled back.

In this driver I tried to simulate transactional behavior similar to one found in ORACLE. The way firebird itself deals with transactions is very different.

I'd like to know you opinion about this driver, design of code, transactional behavior, e.t.c.

TIA

Subject: Re: Firebird
Posted by [mirek](#) on Thu, 30 Jun 2011 06:20:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

So far looks good. Fill the rest of methods and we could add this as plugin/firebird to uppsrc.

I am not quite sure about TRANSACTION macro.

Mirek

Subject: Re: Firebird

Posted by [unodgs](#) on Thu, 30 Jun 2011 06:44:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

TRANSACTION macro looks good. I like it. The only problem is how to add custom code to handle rollback situation? And does your macro breaks immediately after first error or run queries to the last and then do rollback? Normally I use try/catch scenario.

```
Sql q;
try {
    q.Begin()
    ...
    q.Commit()
}
catch(SqlExc e)
{
    q.Rollback();
    ...
}
```

Subject: Re: Firebird

Posted by [Novo](#) on Mon, 04 Jul 2011 04:44:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 30 June 2011 02:20 So far looks good. Fill the rest of methods and we could add this as plugin/firebird to uppsrc.

I am not quite sure about TRANSACTION macro.

Mirek

I added support for almost everything except of blob. I'm also going to develop a bunch of tests for all this stuff.

IMHO, TRANSACTION macro is very convenient to use. You just declare a block of code as running inside of a transaction. No extra variable declarations, no complexity related to exception handling, no worries about jumping out of a block because of business logic.

This macro is similar by design to the INTERLOCKED_ macro. They serve similar needs in different domains.

Subject: Re: Firebird

Posted by [Novo](#) on Mon, 04 Jul 2011 04:50:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 30 June 2011 02:44 TRANSACTION macro looks good. I like it. The only problem is how to add custom code to handle rollback situation? And does your macro breaks immediately after first error or run queries to the last and then do rollback? Normally I use try/catch scenario.

```
Sql q;
try {
    q.Begin()
    ...
    q.Commit()
}
catch(SqlExc e)
{
    q.Rollback();
    ...
}
```

Well, there is no way to add custom rollback code using this macro. But because this macro is very simple, you can implement a similar one using the same pattern. You can take a look at implementation of TRANSACTION_RETAIN in implementation of firebird driver.

B.t.w., your code will look like below.

```
Sql q;

TRANSACTION(q) {
    ...
}
```

You still need to catch exception ...

But what will happen with your code if exception is not of type SqlExc? TRANSACTION will take care of this.

Subject: Re: Firebird
Posted by [Novo](#) on Tue, 05 Jul 2011 04:15:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

I updated firebird driver again. Blob seems to be implemented. A lot of stuff needs tests and there are known issues.

I'm also not absolutely sure about type conversion logic.
I, probably, also need to support ValueGen.
I also found OracleBlob and OracleClob in the Oracle8 driver.

Similar classes can be implemented with firebird. Unfortunately, these classes is not a generic solution. Blobs is a part of almost all databases. So, having separate classes, which implement BlockStream interface for each database, doesn't seem to be a perfect solution.

I will appreciate your feedback.

TIA

Subject: Re: Firebird
Posted by [Novo](#) on Wed, 06 Jul 2011 04:07:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another update.

At this time I made tests for all data types except of Blob.

Can somebody tell me what else is missing, or not working, or not done well?

TIA

Subject: Re: Firebird
Posted by [Novo](#) on Thu, 07 Jul 2011 04:40:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Added support for schema. Fixed compilation on Linux.
Tested Blob. It works fine.
The only missing part is a Script Runner. I'm going to add it after this weekend.

Subject: Re: Firebird
Posted by [Novo](#) on Sat, 09 Jul 2011 04:49:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 30 June 2011 02:44And does your macro breaks immediately after first error or run queries to the last and then do rollback?

TRANSACTION macro will rollback if you don't reach end of a block. This can happen in case of an exception, "break", "return", or "goto" statement. It breaks immediately, the rest of statements in the block won't be executed. "break", "return", and "goto" statements work similar to exception in this case.

Well, this macro is just another trick with scopes.

Subject: Re: Firebird
Posted by [Novo](#) on Sat, 23 Jul 2011 04:22:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another update.

Added tests for blobs. Implemented script runner and several other methods. The only missing functionality is multi-level transactions. Taking in the account different transaction model of Firebird I do not want to mess with it.

I think I'm done with it.

Well, after more intensive testing I found bugs ...

Subject: Re: Firebird
Posted by [Novo](#) on Sun, 31 Jul 2011 17:02:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Update.

Fixed known problems, implemented ValueGen for Firebird, added more tests.

At this point I will switch into bugfixing mode.

Subject: Re: Firebird
Posted by [Novo](#) on Tue, 30 Aug 2011 03:40:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Firebird driver is in bazaar now.
