
Subject: OleDb + parameters = problem
Posted by [Novo](#) on Sat, 17 Jan 2009 17:16:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Code:

```
void Test::remove_person(void)
{
    StringBuffer sql;
    Sql stmt(db_session);

    sql << "DELETE FROM person ";
    sql << "WHERE id = ?";

    try
    {
        stmt.SetStatement(sql);
        stmt.SetParam(0, sql_person(0));

    stmt.ExecuteX();
    }
    catch (const SqlExc& e)
    {
        sql_person.CancelRemove();
        Exclamation("[* " + DeQtLf(e) + "]");
    }
}
```

Problem: Execute(OleDb): "Multiple-step OLE DB operation generated errors. Check each OLE DB status value, if available. No work was done. (Microsoft JET Database Engine)
SQL error: DELETE FROM person WHERE id = ?"

The same code was working fine when I wasn't using parameters, and was substituting value into the SQL statement.

Similar approach with parameters works fine with SQLITE database.

Am I missing something?

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Mon, 19 Jan 2009 09:14:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, I have to admit I am not using parameters mode too much recently; usually it buys you

nothing.

Anyway, out of curiosity, I have tried in my current application to replace some SqlExp with explicit params:

```
#if 1
    sql.SetParam(0, cat.GetKey());
    sql.Execute("select ID, TEXT from ITEM_TYPE where ITEM_CATEGORY_ID = ? order by
TEXT");
#else
    SqlBool wh;
    if(!IsNull(cat.GetKey()))
        wh = ITEM_CATEGORY_ID == cat.GetKey();
    sql * Select(ID, TEXT).From(ITEM_TYPE).Where(wh).OrderBy(TEXT);
#endif
```

In any case, maybe you can try inverse conversion - from SetParam to SqlExp....

BTW, StringBuffer buys you nothing here over plain String and you can simplify your life intensely by using global SQL (as long as you have only single main rdbms):

```
void Test::remove_person(void)
{
    try
    {
        SQL.SetParam(0, sql_person(0));
        SQL.ExecuteX("DELETE FROM person WHERE id = ?");
    }
    catch (const SqlExc& e)
    {
        sql_person.CancelRemove();
        Exclamation("[* " + DeQtflf(e) + "]);
    }
}
```

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Tue, 20 Jan 2009 22:34:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for your answer.

luzr wrote on Mon, 19 January 2009 04:14 Well, I have to admit I am not using parameters mode too much recently; usually it buys you nothing.

I haven't check how SqlExp with explicit params works, but using parameters with SQL is very important. If you aren't using parameters, then server has to parse your SQL all the time. And very often parsing (and creating an execution plan) is more expensive than executing itself.

Quote:

In any case, maybe you can try inverse conversion - from SetParam to SqlExp....

Actually, I'm planing to move from SetParam to my own database code as soon as I will be able to integrate it into TheIDE build system, which is unfortunately very hard to use in workspace-like way.

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Wed, 21 Jan 2009 13:14:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 20 January 2009 17:34

I haven't check how SqlExp with explicit params works, but using parameters with SQL is very important. If you aren't using parameters, then server has to parse your SQL all the time. And very often parsing (and creating an execution plan) is more expensive than executing itself.

Well, have you benchmarked? I have stopped using precompiled Sql expressions when I noticed there is no difference you can really feel.

It might be useful for some very corner cases like when you are querying single value from single table; in all other cases, pushing data over network will nullify any advantage you can gain by keeping your statement prepared.

OTOH, it is still supported. Note also that one nice way how to keep very frequent SQL statements precompiled on the server is to have them static:

```
void Test::remove_person(void)
{
    static Sql stmt("DELETE FROM person WHERE id = ?");
```

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Wed, 21 Jan 2009 13:22:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Speaking about it, it would be really nice if you can provide the benchmark.

In fact, current SqlExp should be possible to expand to support parameters as well, with automatic cache of prepared statements.

Just show me it has any benefit

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Fri, 23 Jan 2009 04:48:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Wed, 21 January 2009 08:22Speaking about it, it would be really nice if you can provide the benchmark.

In fact, current SqlExp should be possible to expand to support parameters as well, with automatic cache of prepared statements.

Just show me it has any benefit

Mirek

http://www.ultimatepp.org/forum/index.php?t=msg&goto=11505&#msg_11541

Try to compare time of insertion of 100000 records. I believe I included drivers for SQLITE and Firebird, which is not supported by U++. I can try to build drivers for other databases and develop more tests. But that will take some time.

From my personal experience I know that using parameters with ORACLE speeds up database operations approximately three times.

Just try to monitor SQL-cache when inserting millions of records not using parameters. Actually, that would be not absolutely correct test because ORACLE is able to auto-detect parameters-capable SQL and transform SQL before executing it.

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Fri, 23 Jan 2009 04:55:28 GMT

luzr wrote on Wed, 21 January 2009 08:14

OTOH, it is still supported. Note also that one nice way how to keep very frequent SQL statements precompiled on the server is to have them static:

```
void Test::remove_person(void)
{
    static Sql stmt("DELETE FROM person WHERE id = ?");
```

Mirek

Precompiled statements get invalidated after you execute a DDL-statement, if I recall that correctly. I think that managing of precompiled statements is more complicated than just declaring a statement static.

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Fri, 23 Jan 2009 05:03:59 GMT

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

Novo wrote on Thu, 22 January 2009 23:48I can try to build drivers for other databases and develop more tests. But that will take some time.

B.T.W. Improving of UPP-build system would help a lot in this task.

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Fri, 23 Jan 2009 10:36:10 GMT

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

Novo wrote on Thu, 22 January 2009 23:55luzr wrote on Wed, 21 January 2009 08:14
OTOH, it is still supported. Note also that one nice way how to keep very frequent SQL statements precompiled on the server is to have them static:

```
void Test::remove_person(void)
{
    static Sql stmt("DELETE FROM person WHERE id = ?");
```

Mirek

Precompiled statements get invalidated after you execute a DDL-statement, if I recall that correctly. I think that managing of precompiled statements is more complicated than just declaring a statement static.

How so? That would mean invalidating secondary cursor, IMO, impossible.

Consider:

```
Sql sql1, sql2;
sql1 * Select(ID).From(DATA).Where(mycondition);
while(sql1.Fetch())
    sql2 * Update(DATA)(FIELD, value).Where(ID == sql1[ID]);
```

Any RDBMS worth of its name must support such loop. And if this loop works, I believe that declaring cursor static should work too (but I might be wrong, of course).

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Fri, 23 Jan 2009 10:52:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, you have been proved right w.r.t. prepared statements and speed, at least with MSSQL:

```
void Benchmark()
{
    Sql sql;
    sql * Select(ID).From(SUBJECT);
    Vector<int> id;
    while(sql.Fetch())
        id.Add(sql[0]);
    for(int i = 0; i < id.GetCount(); i++) {
        RTIMING("Unprepared");
        sql % Select(TEXT).From(SUBJECT_ATTR).Where(ATTR == "NAME" && SUBJECT_ID ==
id[i]);
    }
    sql.SetStatement("select TEXT from SUBJECT_ATTR where ATTR = ? and SUBJECT_ID = ?");
    for(int i = 0; i < id.GetCount(); i++) {
        RTIMING("Prepared");
        sql.Run("NAME", id[i]);
        sql.Fetch();
    }
}
```

TIMING Prepared : 79.99 ms - 273.94 us (80.00 ms / 292), min: 0.00 ns, max: 1.00 ms,
nesting: 1 - 292
TIMING Unprepared : 483.99 ms - 1.66 ms (484.00 ms / 292), min: 0.00 ns, max: 15.00 ms,
nesting: 1 - 292

Well, SqlExp cache will be fun Thanks for makeing me aware about this.

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Fri, 23 Jan 2009 10:59:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Anyway, it seems like DDL does not have such bad effect on precompiled statements after all:

```
void Test()
{
    Sql sql;
    sql * Select(ID).From(SUBJECT);
    Vector<int> id;
    while(sql.Fetch())
        id.Add(sql[0]);
    sql.SetStatement("select TEXT from SUBJECT_ATTR where ATTR = ? and SUBJECT_ID = ?");
    for(int i = 0; i < id.GetCount(); i++) {
        sql.Run("NAME", id[i]);
        if(sql.Fetch()) {
            DDUMP(sql[0]);
            SQL * Update(SUBJECT_ATTR)(ORG_ID, -1).Where(ID == id[i]);
        }
    }
}
```

Still has correct output...

I guess you can use the 'static' without too much worries.

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Sat, 24 Jan 2009 16:58:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Fri, 23 January 2009 05:59 Anyway, it seems like DDL does not have such bad effect on precompiled statements after all:

I checked my code.

In case of DDL statement and:

ADO - (I was using it with JET and MSSQL Server) I commit transaction and clear statement cache.

SQLITE - clear statement cache and finalize current statement (I'm using old API).

Interbase/Firebird - clear statement cache, rollback transaction, execute DDL, commit transaction. (This database has slightly different transaction model).

this is a bug, or it was tested only with MS SQL Server.

ORACLE - clear statement cache, execute DDL, mark all open statements as invalidated by DDL (they should be prepared again).

Microsoft has different APIs to work with MS SQL Server. Prepare-Bind-Execute model is implemented using special stored procedures, which are often called by a high-level API. You can check that in FreeTDS ODBC driver. It is open-source.

Subject: Re: OleDb + parameters = problem
Posted by [mirek](#) on Sat, 24 Jan 2009 19:19:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Sat, 24 January 2009 11:58 luzr wrote on Fri, 23 January 2009 05:59 Anyway, it seems like DDL does not have such bad effect on precompiled statements after all:

I checked my code.

In case of DDL statement and:

ADO - (I was using it with JET and MSSQL Server) I commit transaction and clear statement cache.

SQLITE - clear statement cache and finalize current statement (I'm using old API).

Interbase/Firebird - clear statement cache, rollback transaction, execute DDL, commit transaction. (This database has slightly different transaction model).

this is a bug, or it was tested only with MS SQL Server.

ORACLE - clear statement cache, execute DDL, mark all open statements as invalidated by DDL (they should be prepared again).

Microsoft has different APIs to work with MS SQL Server. Prepare-Bind-Execute model is implemented using special stored procedures, which are often called by a high-level API. You can check that in FreeTDS ODBC driver. It is open-source.

Well, sorry, my mistake, I have messed DDL with DML...

Maybe that why I have not ever encountered the problem in practice.

Seriously, do you often issue DDL during the application run?

IMO, in any multiuser environment, such thing is extremely unlikely because regular users do not have rights to do DDL.

OTOH, if I ever add automated prepared statement cache with SqlExp, "Invalidate" seems like a good idea, right?

Mirek

Mirek

Subject: Re: OleDb + parameters = problem
Posted by [Novo](#) on Sun, 25 Jan 2009 03:44:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Sat, 24 January 2009 14:19
Seriously, do you often issue DDL during the application run?

Happens all the time in my unit-tests.

Seriously, such situation is possible when you create a database schema and populate it with initial data.

Everything, what is theoretically possible, will be done by an user.

Quote:

OTOH, if I ever add automated prepared statement cache with SqlExp, "Invalidate" seems like a good idea, right?

That's right. Actually, cache of prepared statements has sense only in case of using parameters, otherwise there is no reason to reuse statements.
