

---

Subject: Re: ODBC Assertion failed

Posted by [Giorgio](#) on Wed, 27 Jan 2021 15:44:21 GMT

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

---

Hello,

I made a lot of tests in the last weeks and I still cannot wrap my head around this error.

To begin with, I had some issues with DDUMP: if I put that before the assert and I try to compile in release mode I got "error C2018: unknown character '0x40'". So I used RDUMP in release. In debug mode DDUMP is ok.

Anyway, I noticed in the log "tlevel = 0" and just after that "tlevel = -1" so inspecting closely the code I found this in the destructor of the class:

```
if(my_mssql_db.IsOpen()){
    my_mssql_db.Commit();
    my_mssql_db.Close();
}
```

Actually, that extra .Commit() was wrong (there was no matching between the .Begin and the .Commit/.Rollback) and after I removed it the "tlevel = -1" disappeared... but the application crashed anyway.

So I tried something different. The code I posted initially is actually a simplified version of my code. What I'm doing is inserting a header/body document, so I have a query that insert the header in a table and after that a for cycle for the body in a different table. Initially I used just one Sql object both for the header and the body, so I tried to use two different Sql objects, one for the body and one for the header:

```
bool InsertDocument(ValueMap * header, vector<vm*> * rows){
```

```
    Sql query_header(my_mssql_db);
    query_header.ClearError();
```

```
    Sql query_body(my_mssql_db);
    query_body.ClearError();
```

```
    query_header.Begin();
    query_body.Begin();
```

```
    try{
        query_heder * Insert(My_mssql_header_table)(*header);
    } catch(SqlExc) {
        ErrorOK(query_header.GetLastError());
    }
}
```

```

for(vector<vm*>::iterator it = rows->begin(); it != rows->end(); ++it) {
    try{
        query_body * Insert(My_mssql_body_table)(*it);
    } catch(SqlExc) {
        ErrorOK(query_body.GetLastError());
    }
}

if(query_header.WasError() || query_body.WasError()){
    query_body.Rollback();
    query_header.Rollback();
    return false;
}

query_header.Commit();
query_body.Commit();
return true;
}

```

After a couple of tests it seemed to work, but in the end also this approach led to crashes. At the end I decided to scrap the Commit altogether, so each insert it's by itself. After a couple of week of usage I got no crashes. This approach can lead to some inconsistencies (e.g. an header without a body), but luckily enough this is a quite isolated part of the application, so I can manage manually this kind of situation.

What baffles me is that - AFAIK - using a transaction to bundle together a bunch of insert is perfectly fine (it's actually the best solution because in case of problem you rollback the entire set of instructions - both the insert in the header and those in the body), so I cannot understand why using "isolated" insert is working while Commit leads to crashes.

Regards,  
gio