

---

Subject: How do I load sqlite3 .sql with triggers on it?  
Posted by [rxantos](#) on Fri, 31 Aug 2012 13:38:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

How do I load a .sql with triggers?

I tried to load the .sql to a String and then use `Sqlite3PerformScript`;

I also tried `SQL.Execute`, but that only executes one line.

`Sqlite3PerformScript` runs until it gets to statements like this:

```
create trigger if not exists ti_usuario after insert on USUARIO begin
  update USUARIO set UTIME = datetime('now') where ID = new.ID;
end;
```

and gives me an error like this:

```
ERROR near "ID": syntax error(0): Preparing: create trigger if not exists ti_usuario after insert on
USUARIO begin
  update USUARIO set UTIME = datetime('now') where ID = new.ID
```

The same .sql compiles flawlessly to a db when using the `sqlite3` program.

What call shall I use to load a .sql directly?

Update:

It seems that the error is caused because `Sqlite3PerformScript` dives the trigger statement into two. Eating the ';' in the process.

Note that a trigger is ended by a `"; end;"` and not by a simple `;"`. Thus its a single statement.

[http://www.sqlite.org/lang\\_createtrigger.html](http://www.sqlite.org/lang_createtrigger.html)

The following is a work around hack based on `Sqlite3PerformScript`.

Hack, because it will just work for statements started with `"create trigger "`

Updated the hack.

Now it needs `"create trigger "` at the start of the line and `"end;"` on its own line. That way multi statement triggers are possible. Still a hack though.

```
bool Sqlite3PerformScript_hack(const String& txt, StatementExecutor& se, Gate2<int, int>
progress_canceled)
{
```

```

const char* text = txt;
const char * p;
static const char szCreateTrigger[] = "create trigger ";
static const size_t szCreateTrigger_len = sizeof(szCreateTrigger) - 1;

for(;*text; text++) {
    String stmt;
    bool isTrigger = false;

    while(*text <= ' ' && *text > '\0') {
        text++;
    }

    if('\0' == *text) {
        break;
    }

    isTrigger = (0 == _strnicmp(szCreateTrigger,text, szCreateTrigger_len));

    for(;*text;) {
        if('; ' == *text) {
            if(isTrigger) {
                p = text - 3;
                if(!_strnicmp("end;",p,4)) {
                    --p;
                    if((32 >= *p)||(' ' == *p)) {
                        break;
                    }
                }
            }
            stmt.Cat(*text++);
            continue;
        }
        break;
    } else if('\n' == *text) {
        text = Upp::Sqlite3ReadString(text, stmt);
    } else if('\r' == *text) {
        text = Upp::Sqlite3ReadString(text, stmt);
    } else {
        stmt.Cat(*text++);
    }
}

if(progress_canceled(text - txt.Begin(), txt.GetLength())) {
    return false;
}

if(!se.Execute(stmt)) {
    return false;
}

```

```
}  
}  
  
return true;  
}
```

---