
Subject: Prefix to prevent name clash in SQL: Sqlld, variables, tables, ...

Posted by [mingodad](#) on Sun, 17 Feb 2008 12:40:01 GMT

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

Today I was converting one application to upp 2008.1 and found that the actual system of parse database schema can generate name clash:

```
TABLE_(CURRENCIES)
INT_(CURRENCY)
END_TABLE
```

The above will generate a name clash:

main.cpp

In file included from d:\upp\uppsrc/Sql/sch_model.h:162,
from d:\upp\uppsrc/plugin/sqlite3/Sqlite3Schema.h:82,
from d:\upp\uppsrc/Sql/sch_header.h:45,
from d:\MyApps\DADBiz\DADBiz.h:17,
from d:\MyApps\DADBiz\main.cpp:1:

d:\MyApps\DADBiz\dadbiz.sch:33: error: 'Upp::Sqlld CURRENCY' redeclared as different kind of symbol

d:\upp\mingw\include/oidl.h:113: error: previous declaration of 'typedef union CY CURRENCY'

In file included from d:\upp\uppsrc/Sql/sch_model.h:162,
from d:\upp\uppsrc/plugin/sqlite3/Sqlite3Schema.h:82,
from d:\upp\uppsrc/Sql/sch_source.h:6,
from d:\MyApps\DADBiz\DADBiz.h:18,
from d:\MyApps\DADBiz\main.cpp:1:

d:\MyApps\DADBiz\dadbiz.sch:33: error: 'Upp::Sqlld CURRENCY' redeclared as different kind of symbol

d:\upp\mingw\include/oidl.h:113: error: previous declaration of 'typedef union CY CURRENCY'

[FEATURE REQUEST]

So I studied the source code and end up with some modifications that allow us to specify a prefix to the generated symbols with a definition of one preprocessor macro:

```
#define SCHEMA_PREFIX_CPP _ANY_PREFIX_YOU_LIKE_
```

And after that any reference to tables or fields should be done as:

```
dbList.AddKey(_ANY_PREFIX_YOU_LIKE_CURRENCY);
dbList.SetTable(_ANY_PREFIX_YOU_LIKE_CURRENCIES);
dbList.AddColumn(_ANY_PREFIX_YOU_LIKE_CURRENCY, t_("Id"), .SetFormat("%8>d");
dbList.SetOrderBy(_ANY_PREFIX_YOU_LIKE_CURRENCY).Limit(100);
```

It doesn't break any application that already works, because without defining SCHEMA_PREFIX_CPP all generated symbols remain as before.

In this process I've found a bug in GCC when joining string in macros, in some cases it adds a

space between the tokens, that's why the macro `ADD_SCHEMA_PREFIX_CPP2` was created.

I've sending attached the files I've modified.

I'll hope it will be usefull to others and probably be included in the official distribution too !

Thanks for all of you that help develop this fantastic tool !

If you need more explanations, ask me !

Ps:

[FEATURE REQUEST]

As well to allow applications made with `upp` to work with databases, tables generated by other programs that doesn't create table and record fields names in uppercase (commonly lowercase) a patch to `sql.cpp` to make comparisons after convert field names retrieved fron queries to upper case.

Instead of:

```
// sql.cpp line 254
Value Sql::operator[](SqlId id) const {
    String s = ~id;
    for(int i = 0; i < cn->info.GetCount(); i++)
        if(cn->info[i].name == s)
            return operator[](i);
    return Value();
}
```

Use this :

```
Value Sql::operator[](SqlId id) const {
    String s = ~id;
    for(int i = 0; i < cn->info.GetCount(); i++)
    {
        String sb = ToUpper(cn->info[i].name);
        if(sb == s)
            return operator[](i);
    }
    return Value();
}
```

File Attachments

- 1) [sch_header.h](#), downloaded 553 times
 - 2) [sch_model.h](#), downloaded 597 times
 - 3) [sch_source.h](#), downloaded 847 times
 - 4) [Sql.cpp](#), downloaded 655 times
-