
Subject: template + convert problem

Posted by [sergeynikitin](#) on Sat, 19 Sep 2009 12:14:55 GMT

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

I make a table in which multiple fields of reference from other tables. The mapping of the field from another table I made with the converter.

Now it looks as follows:

converts.h

```
struct ConvCompany : Convert
{
    virtual Value Format(const Value& q) const;
};
```

```
struct ConvNomenc1 : Convert
{
    virtual Value Format(const Value& q) const;
};
```

```
struct ConvManager : Convert
{
    virtual Value Format(const Value& q) const;
};
```

converts.cpp:

```
Value ConvCompany::Format(const Value &q) const
{
    if(q.IsNull()) return Null;
    static VectorMap<int, String> comp;
    static Time lastcleartime;
    if(GetSysTime()-lastcleartime > 600){
        comp.Clear();
        lastcleartime=GetSysTime();
    }
    int f = comp.Find(int(q));
    if(f >= 0){
        return comp[f];
    } else {
        Sql sql;
        sql * Select(COM_NAME).From(COMpany).Where(COM_ID == q);
        String company;
        if(sql.Fetch()) {
            int sid = int(q);
```

```

    company = sql[COM_NAME];
    comp.Add(sid, company);
} else {
    company = "";
}
return company;
}
}
Value ConvNomencl::Format(const Value& q) const
{
    if(q.IsNull()) return Null;
    static VectorMap<int, String> nom;
    static Time lastcleartime;
    if(GetSysTime()-lastcleartime > 600){
        nom.Clear();
        lastcleartime=GetSysTime();
    }
    int f = nom.Find(int(q));
    if(f >= 0){
        return nom[f];
    } else {
        Sql sql;
        sql * SqlSelect(NOM_ID,NOM_NAME).From(NOMENCL).Where(NOM_ID == q);
        String nomencl;
        if(sql.Fetch()) {
            int sid = sql[NOM_ID];
            nomencl = sql[NOM_NAME];
            nom.Add(sid, nomencl);
        } else {
            nomencl = "";
        }
        return nomencl;
    }
}

```

```

Value ConvManager::Format(const Value &q) const
{
    if(q.IsNull()) return Null;
    static VectorMap<int, String> man;
    static Time lastcleartime;
    if(GetSysTime()-lastcleartime > 600){
        man.Clear();
        lastcleartime=GetSysTime();
    }
    int f = man.Find(int(q));
    if(f >= 0){
        return man[f];
    } else {

```

```

Sql sql;
sql * SqlSelect(USR_REALNAME).From(USER).Where(USR_ID == q);
String manager;
if(sql.Fetch()) {
    int sid = int(q);
    manager = sql[USR_REALNAME];
    man.Add(sid, manager);
} else {
    manager = "";
}
return manager;
}
}

```

These converters differ only in the name of the table and field names. I want this to be in the form of a template, something like:

```

template <SqlId TBL ,SqlId ID, SqlId NAME,int CLEARTIME>
struct ConvDict : Convert
//ConvDict<COMPANY,COM_ID,COM_NAME,600>()
{
    virtual Value Format(const Value& q) const {
        if(q.IsNull()) return Null;
        static VectorMap<int, String> dict;
        static Time lastcleartime;
        if(GetSysTime()-lastcleartime > CLEARTIME){
            dict.Clear();
            lastcleartime=GetSysTime();
        }
        int f = dict.Find(int(q));
        if(f >= 0){
            return dict[f];
        } else {
            Sql sql;
            sql * Select(NAME).From(TBL).Where(ID == q);
            String s;
            if(sql.Fetch()) {
                int sid = int(q);
                s = sql[NAME];
                dict.Add(sid, s);
            } else {
                s = "";
            }
            return s;
        }
    }
};

```

But this code is not compiled, wrote that at this point can not be applied Sqlld.

Can someone experienced with this?

How do make a template for the Converter, or Display?

Subject: Re: template + convert problem

Posted by [Didier](#) on Sun, 20 Sep 2009 09:56:50 GMT

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

Hi sergey,

The only words template declaration accepts are: 'class' (or typename) and 'int', all other notations are false and will not compile.

class : means anything, any variable, constant, ... whatever you want

int : means only literal integer numbers (1, 2, 1110, 0x10, ...) : no variables !!

But in you're case try this

```
template <class TBL ,class ID, class NAME,int CLEARTIME>
struct ConvDict : Convert
{
    virtual Value Format(const Value& q) const {
        if(q.IsNull()) return Null;
        static VectorMap<int, String> dict;
        static Time lastcleartime;
        if(GetSysTime()-lastcleartime > CLEARTIME){
            dict.Clear();
            lastcleartime=GetSysTime();
        }
        int f = dict.Find(int(q));
        if(f >= 0){
            return dict[f];
        } else {
            Sql sql;
            sql * Select(NAME).From(TBL).Where(ID == q);
            String s;
            if(sql.Fetch()) {
                int sid = int(q);
                s = sql[NAME];
                dict.Add(sid, s);
            } else {
```

```

    s = "";
}
return s;
}
}
};

```

I haven't tried it, but at least the declaration is correct

Subject: Re: template + convert problem
 Posted by [Didier](#) on Sun, 20 Sep 2009 10:08:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

After another look, something is missing. You mixed up type passing (template paremeters), and specific value fixing (you're parameters).

You need a constructor, no need for template !

The following code compiles:

```

struct ConvDict : Convert
{
private:
const SqlId TBL;
const SqlId ID;
const SqlId NAME;
const int  CLEARTIME;

public:
ConvDict(SqlId tbl ,SqlId id, SqlId name,int clearTime)
: TBL(tbl)
, ID(id)
, NAME(name)
, CLEARTIME(clearTime)
{

}
}

```

```

virtual Value Format(const Value& q) const
{
    if(q.IsNull()) return Null;
    static VectorMap<int, String> dict;
    static Time lastcleartime;
    if(GetSysTime()-lastcleartime > CLEARTIME)
    {
        dict.Clear();
        lastcleartime=GetSysTime();
    }

    int f = dict.Find(int(q));
    if(f >= 0)
    {
        return dict[f];
    } else {
        Sql sql;
        sql * Select(NAME).From(TBL).Where(ID == q);
        String s;
        if(sql.Fetch()) {
            int sid = int(q);
            s = sql[NAME];
            dict.Add(sid, s);
        } else {
            s = "";
        }
        return s;
    }
}
};

```