
Subject: Syntactic sugar
Posted by [unodgs](#) on Mon, 20 Oct 2008 12:22:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

I think it would be nice to add to Sqlld interface overloaded [] operator:
SqlCol operator [] (const Sqlld& id) const;
which let us to write

```
Select(ID, CUSTOMER[NAME], PRODUCT[QUANTITY])...
```

insead of

```
Select(ID, NAME.Of(CUSTOMER), QUANTITY.Of(PRODUCT))...
```

What do you think?

Subject: Re: Syntactic sugar
Posted by [unodgs](#) on Mon, 20 Oct 2008 12:33:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

We could also add extended version of TABLE macro with alias parameter.

```
TABLE(CUSTOMER, CU)  
END_TABLE
```

```
Select(CU[NAME], CU[AGE])  
.From(CUSTOMER).As(CU) //of course As is not necessary here
```

Subject: Re: Syntactic sugar
Posted by [sergeynikitin](#) on Mon, 20 Oct 2008 13:28:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is very good. When did it happen?

By the way. There is a tool - Clarion, - to create applications under Win.

There is construction
Nametable Table,Pre(Nam)

and further in all fields table Nametable differ from the fields of other tables prefix Nam - it looks like this:

Nam:ID

or

Nam:Comment

The design is very convenient. I wish that the UPP would be such a structure.

Subject: Re: Syntactic sugar

Posted by [unodgs](#) on Mon, 20 Oct 2008 19:26:31 GMT

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

Try this :

```
#include <Core/Core.h>
```

```
#include <Sql/Sql.h>
```

```
using namespace Upp;
```

```
struct CustomerTable
```

```
{
```

```
    const SqlId ID;
```

```
    const SqlId NAME;
```

```
    const SqlId AGE;
```

```
operator SqlId()
```

```
{
```

```
    return SqlId("CUSTOMER");
```

```
}
```

```
CustomerTable() :
```

```
    ID("CUSTOMER.ID"),
```

```
    NAME("CUSTOMER.NAME"),
```

```
    AGE("CUSTOMER.AGE")
```

```
    {}
```

```
};
```

```
CustomerTable CUSTOMER;
```

```
#define CUS CUSTOMER //alias
```

```
CONSOLE_APP_MAIN
```

```
{
```

```
    SqlSelect s =
```

```
        ::Select(CUSTOMER.NAME, CUSTOMER.AGE)
```

```
        .From(CUSTOMER)
```

```
        .Where(CUS.ID == 10);
```

```
SqlStatement stmt = s;  
Cout() << stmt.GetText() << '\n';  
}
```

If we expand TABLE macros in this way it would be possible to use natural SQL field qualification. If you would like to use :: instead dot you should declare all Sqlld's inside the CustomerTable as static, but this is a more complicated way and there is a problem with From. Anyway I think I'll explore my new approach

Subject: Re: Syntactic sugar
Posted by [sergeynikitin](#) on Mon, 20 Oct 2008 20:04:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Doesn't matter what sign used in expression '.' or ':' .

Any case this form is better then FIELD.Of(TABLE)

How soon it will appear in a working version? Already want to write new .

Subject: Re: Syntactic sugar
Posted by [sergeynikitin](#) on Mon, 20 Oct 2008 20:18:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another small question. How to show in single SqlArray fields from linked tables (as a Join)?

Subject: Re: Syntactic sugar
Posted by [mirek](#) on Fri, 31 Oct 2008 13:52:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Mon, 20 October 2008 08:33We could also add extended version of TABLE macro with alias parameter.

```
TABLE(CUSTOMER, CU)  
END_TABLE
```

```
Select(CU[NAME], CU[AGE])  
.From(CUSTOMER).As(CU) //of course As is not necessary here
```

I am afraid this would be prone to collide with TABLE inheritance....

Mirek

Subject: Re: Syntactic sugar

Posted by [mirek](#) on Fri, 31 Oct 2008 13:54:23 GMT

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

sergeynikitin wrote on Mon, 20 October 2008 16:18 Another small question. How to show in single SqlArray fields from linked tables (as a Join)?

Well, the most straightforward is not to use SqlArray

What you want requires select from multiple tables. SqlArray is designed to support only one table.

Anyway, SqlArray has Join to create master-detail relation of two SqlArrays (or, in fact, you can Join SqlArray to ArrayCtrl too).

Mirek

Subject: Re: Syntactic sugar

Posted by [mirek](#) on Fri, 31 Oct 2008 14:02:18 GMT

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

unodgs wrote on Mon, 20 October 2008 08:22 I think it would be nice to add to SqlId interface overloaded [] operator:

SqlCol operator [] (const SqlId& id) const;
which let us to write

Select(ID, CUSTOMER[NAME], PRODUCT[QUANTITY])...

insead of

Select(ID, NAME.Of(CUSTOMER), QUANTITY.Of(PRODUCT))...

What do you think?

Well, it felt a bit weird at first, but after a bit of thinking, I think this is quite natural.

Added. (And sorry for the delay, T++/A++ is really driving me insane

Mirek
