Subject: How to use Sql * Update
Posted by michael on Wed, 24 Dec 2008 13:47:35 GMT
View Forum Message <> Reply to Message

I'm trying to use the following code to update a sql-table:

```
Sql sql(session);

sql * Update(SqlId(sqlTable))
(NAME, customer.name)
(NUMMER, customer.number)
(PRODUKT, customer.product)
(VERSION, customer.version)
(STANDORT, customer.location)
(ANBINDUNG, customer.connection)
(BEMERKUNG, customer.comment)
.Where(ID == customer.sqlID);
```

I get these error:

```
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1274) : error C2660: 'Upp::Ctrl::Update' : function does not take 1 arguments
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1282) : error C2228: left of '.Where' must have class/struct/union
prohibisZA: 1 file(s) built in (0:01.87), 1872 msecs / file, duration = 1919 msecs, parallelization 0%

There were errors. (0:02.48)
```

What went wrong?

Subject: Re: How to use Sql * Update
Posted by michael on Wed, 24 Dec 2008 16:45:29 GMT
View Forum Message <> Reply to Message

Found out that this works:

```
Sql sql(session);

sql * SqlUpdate(SqlId(sqlTable))
(NAME, customer.name)
(NUMMER, customer.number)
(PRODUKT, customer.product)
(VERSION, customer.version)
(STANDORT, customer.location)
(ANBINDUNG, customer.connection)
(BEMERKUNG, customer.comment)
```

.Where(ID == customer.sqlID);

So is it ok to use SqlSelect, SqlInsert and SqlUpdate instead of Select, Insert and Update(which does not work)?

Subject: Re: How to use Sql * Update
Posted by mirek on Thu, 25 Dec 2008 20:35:56 GMT
View Forum Message <> Reply to Message

Well, there is unfotunate nameclash with Update that usually strikes when doing SQL in GUI - Ctrl has Update method, so that takes preference.

Update and SqlUpdate are equivalent.

Mirek

Subject: Re: How to use Sql * Update
Posted by michael on Fri, 26 Dec 2008 10:32:29 GMT
View Forum Message <> Reply to Message

But SqlSelect isn't equivalent to Select?

When i do this:

sql * Select(SqlId(sqlWhat)).From(SqlId(sqlTable)).Where(sqlWhere).OrderBy(sqlOrder);
Everthing is ok, but when i do this:

sql * SqlSelect(SqlId(sqlWhat)).From(SqlId(sqlTable)).Where(sqlWhere).OrderBy(sqlOrder)
I get the following error:

H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2440: '<function-style-cast>' : cannot convert from 'Upp::SqlId' to 'Upp::SqlS
elect'
No constructor could take the source type, or constructor overload resolution was ambiguous
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.From' must have class/struct/union
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.Where' must have class/struct/union
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.OrderBy' must have class/struct/union
prohibisZA: 1 file(s) built in (0:01.96), 1966 msecs / file, duration = 1997 msecs, parallelization 0%

There were errors. (0:02.63)

Subject: Re: How to use Sql * Update
Posted by mirek on Wed, 21 Jan 2009 13:07:50 GMT
View Forum Message <> Reply to Message

michael wrote on Fri, 26 December 2008 05:32But SqlSelect isn't equivalent to Select?

When i do this:

sql * Select(SqlId(sqlWhat)).From(SqlId(sqlTable)).Where(sqlWhere).OrderBy(sqlOrder);
Everthing is ok, but when i do this:

sql * SqlSelect(SqlId(sqlWhat)).From(SqlId(sqlTable)).Where(sqlWhere).OrderBy(sqlOrder)
I get the following error:

H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2440: '<function-style-cast>' : cannot convert from 'Upp::SqlId' to 'Upp::SqlS
elect'
No constructor could take the source type, or constructor overload resolution was ambiguous
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.From' must have class/struct/union
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.Where' must have class/struct/union
H:\Entwicklung\UPP\prohibisZA\prohibisZA.cpp(1069) : error C2228: left of '.OrderBy' must have class/struct/union
prohibisZA: 1 file(s) built in (0:01.96), 1966 msecs / file, duration = 1997 msecs, parallelization 0%

There were errors. (0:02.63)

Yes, but it was not intentional, rather forgotten. Now fixed.