

I think you should get your Values right....

```
StrDbl(arrayLineItems.GetColumn(i,2).ToString())
```

This is really not how we meant it :)

In general, GetColumn returns Value and if you have put double into ArrayCtrl or SqlArray, it will return double Value back. So normally above should be something like

```
(double)arrayLineItems.GetColumn(i,2);
```

Also

```
SQL * Select(CUST_ID, CUSTNAME).From(CUSTOMERS);  
while (SQL.Fetch())  
{  
    cbCustomers.Add(~SQL[CUST_ID], ~SQL[CUSTNAME]);  
}
```

operator~ in this context converts perfectly good typed Value into String. Probably not something I would do to CUST_ID.

```
if (txtPrice.GetData().IsNull()) return;
```

You can just write IsNull(txtPrice). Instead of GetData, consider using operator~ (that in U++ has general meaning something like "other value representation").

```
arrayLineItems.AddColumn("Description", 80);  
....  
(DESCRIPTION, arrayLineItems.GetColumn(i,1).ToString())
```

Sometimes it is worth providing IDs even if you are not directly dealing with SQL, leads to cleaner code

```
arrayLineItems.AddColumn(DESCRIPTION, "Description", 80);
```

```
....  
(DESCRIPTION, arrayLineItems.GetColumn(i,DESCRIPTION))
```

(ToString here is not needed as well as AFAIK it is already String).

```
SQL * Select(TAXABLE).From(CUSTOMERS).Where(CUST_ID == idNum);  
SQL.Fetch();  
optCustTaxable.Set(SQL[0]);
```

As this is pretty common code, there is a shortcut:

```
optCustTaxable.Set(SQL % Select(TAXABLE).From(CUSTOMERS).Where(CUST_ID ==  
idNum));
```

Mirek
