
Subject: Re: sqlarray and sqlite

Posted by [jimlef](#) on Thu, 24 Sep 2020 04:08:03 GMT

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

I may have fixed my own problem, and found my mistake... in my code, I was using sql statements to generate the global SQL in the customers module. I needed to move those to the main line. So here's the new code for anyone looking:

```
#main.cpp
#include "Invoices.h"

#define MODEL "Invoices/Tables/Invoices.sch"
#include <Sql/sch_schema.h>
#undef MODEL

void Invoices::MainMenu(Bar& bar)
{
    bar.Add(t_("Customers"), [=]{ if(!cwin.IsOpen()) cwin.Open(this); });

    // bar.Add(t_("Products"), [=]{ if(!prodwin.IsOpen()) prodwin.Open(this); });

    // bar.Sub(t_("Transactions"), THISFN(TransactionsMenu));
    // bar.Sub(t_("Reports"), THISFN(ReportsMenu));
    bar.Sub(t_("Management"), THISFN(ManagementMenu));
    bar.Add(t_("Exit"), THISFN(Close));
}
/*
void Invoices::TransactionsMenu(Bar& bar)
{
    bar.Add(t_("Create Invoice"), [=]{ if(!createinvoicewin.IsOpen())
    createinvoicewin.Open(this); });
    bar.Add(t_("List Invoices"), [=]{ if(!listinvoiceswin.IsOpen())
    listinvoiceswin.Open(this); });
    bar.Add(t_("List Line Items"), [=]{ if(!listlineitemswin.IsOpen())
    listlineitemswin.Open(this); });
}

void Invoices::ReportsMenu(Bar& bar)
{
    bar.Add(t_("Income / Sales Tax"), [=]{ if(!taxreportwin.IsOpen()) taxreportwin.Open(this); });
    bar.Add(t_("Income / Sales Tax by Customer"), [=]{ if(!taxbycustomerwin.IsOpen())
    taxbycustomerwin.Open(this); });
    bar.Add(t_("Income by Customer"), [=]{ if(!incomewin.IsOpen()) incomewin.Open(this); });
    bar.Add(t_("Profit / Loss"), [=]{ if(!profitwin.IsOpen()) profitwin.Open(this); });
}
*/
void Invoices::ManagementMenu(Bar& bar)
```

```

{
bar.Add(t_("Select Database File"), [=]{myConfig.DBFile = myConfig.SelectDB(); });
bar.Add(t_("Show Current Database File"), [=]{PromptOK(DeQtf("Current Database file is: \n" +
myConfig.DBFile));});
// bar.Add(t_("Set Company Info"), [=]{if(!setcompanywin.IsOpen()) setcompanywin.Open(this); });
bar.Add(t_("Select Output Directory"), [=]{ myConfig.OutputDirectory =
myConfig.GetOutputDirectory(); });
}

```

```
Invoices::Invoices()
```

```

{
CtrlLayout(*this, "Invoices");
AddFrame(mainmenu);
mainmenu.Set(THISFN(MainMenu));

```

```
// from http://leonardoce.interfree.it/leowiki.html "simple configfile"
```

```

if(FileExists(myConfig.configfile))
{
    VectorMap<String, String> cfg = LoadIniFile(myConfig.configfile);
    myConfig.DBFile = cfg.Get("DBFile", Null);
}
else {
    myConfig.GetOutputDirectory();
    myConfig.DBFile = myConfig.SelectDB();
}
}

```

```
// Shortcut keys defined
```

```

/*
bool Invoices::Key(dword key, int count)
{
if(key == K_CTRL_E && !ewin.IsOpen())
    ewin.Open(this);
else
if(key == K_CTRL_F && !fwin.IsOpen())
    fwin.Open(this);
return false;
}
*/

```

```
GUI_APP_MAIN
```

```

{
Configs myConfig;
if(FileExists(myConfig.configfile))
{
    VectorMap<String, String> cfg = LoadIniFile(myConfig.configfile);
    myConfig.DBFile = cfg.Get("DBFile", Null);
}
}

```

```

}
else {
    myConfig.DBFile = myConfig.SelectDB();
}

Sqlite3Session sqlite3;
if(!sqlite3.Open("/home/james/upp/MyApps/Customers/sample.db")) {
    Exclamation("Can't create or open database file\n");
    return;
}

SQL = sqlite3;

Invoices().Run();
}

#Invoices.h
#ifndef __Invoices__Invoices_h_
#define __Invoices__Invoices_h_

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#include <Invoices/UI/sqlincludes.h>
#include "customers.h"
#include "../configs.h"

// Just a quick info dialog showing currently selected file
struct ShowDBWindow : WithShowDBWindowLayout<TopWindow> {
    ShowDBWindow() { CtrlLayout(*this, "Show Current Database File"); }
};

class Invoices : public WithInvoicesLayout<TopWindow> {
    CustomersWindow cwin;
    MenuBar mainmenu;

public:
    typedef Invoices CLASSNAME;
    Configs myConfig;

    Invoices();

    // Menu definition
    void MainMenu(Bar& bar);

    // void TransactionsMenu(Bar& bar);
    // void ReportsMenu(Bar& bar);

```

```

void ManagementMenu(Bar& bar);

// Keys for shortcuts
// bool Key(dword key, int count) override;
};
#endif

#customers.cpp
#include "customers.h"
#define MODEL <Invoices/Tables/Invoices.sch>
#include <Sql/sch_source.h>

#undef MODEL
class AddCustomer : public WithCustomerAddLayout<TopWindow> {
public:
    SqlCtrls ctrls;
    typedef AddCustomer CLASSNAME;
    AddCustomer();
};

/*
ITEM(Option, chkTaxable, SetLabel(t_("Taxable?")).SetFont(SansSerifZ(16)).LeftPosZ(228,
108).TopPosZ(36, 16))
ITEM(EditString, txtCustSearch, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(404, 19))
ITEM(EditString, txtCustName, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(76, 19))
ITEM(EditString, txtCustEmail, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(116, 19))
ITEM(EditString, txtCustPhone, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(156, 19))
ITEM(EditString, txtCustAddress,
SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136, 256).TopPosZ(196, 19))
ITEM(EditString, txtCustCity, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(232, 19))
ITEM(EditString, txtCustState, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
256).TopPosZ(272, 19))
ITEM(EditString, txtCustZip,
MaxChars(10).SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(136,
64).TopPosZ(308, 19))

STRING_ (CUSTNAME, 150) NOT_NULL
STRING_ (EMAIL, 150)
STRING_ (CONTACT, 150)
STRING_ (ADDRESS, 150)
STRING_ (CITY, 50)
STRING_ (STATE, 20)
STRING_ (ZIP, 10)

```

```
INT_ (TAXABLE) NOT_NULL
*/
```

```
AddCustomer::AddCustomer()
{
    CtrlLayoutOKCancel(*this, "Add Customer");
    ctrls
    (CUSTNAME, txtCustName)
    (EMAIL, txtCustEmail)
    (CONTACT, txtCustPhone)
    (ADDRESS, txtCustAddress)
    (CITY, txtCustCity)
    (STATE, txtCustState)
    (ZIP, txtCustZip)
    (TAXABLE, chkTaxable)
;
}
```

```
CustomersWindow::CustomersWindow() {
    CtrlLayout(*this, "Customers");
    // Major assistance on the following code from
    btnNewCustomer << [=] { AddNewCustomer(); }; // THISBACK is not needed in c++11 world and
    could be replaced with lambda.
    /*
    btnSearchCustomer << [=] { btnSearchCustomerClick(); }; // assisted by forum user Klugier
    btnUpdateCustomer << [=] { btnUpdateCustomerClick(); };
    */
```

```
CustArray.SetTable(CUSTOMERS, CUST_ID);
```

```
// CustArray.Join(BOOK_ID, book); // joins id from other db to this id
CustArray.AddColumn(CUSTNAME, "Name"); // .SetConvert(DateIntConvert());
CustArray.AddColumn(EMAIL, "Email"); // .SetConvert(DateIntConvert());
CustArray.AddColumn(CONTACT, "Phone");
CustArray.AddColumn(ADDRESS, "Address");
CustArray.AddColumn(CITY, "City");
CustArray.AddColumn(STATE, "State");
CustArray.AddColumn(ZIP, "Zip");
CustArray.AddColumn(TAXABLE, "Taxable?");
CustArray.ColumnWidths("40 40 20 50 20 15 10 5");
CustArray.NoRemoving();
CustArray.SetOrderBy(CUST_ID);
```

```
CustArray.Query();
```

```
CustArray.WhenLeftDouble = [=] { EditRow(); };
```

```

}

void CustomersWindow::EditRow()
{
    int idNum;
    if(!CustArray.IsCursor())
        return;
    idNum = CustArray.GetKey();
    if (IsNull(idNum))
        return;
    AddCustomer dlg;
    dlg.Title("Edit Customer");

    //if(!dlg.ctrls.Load(CUSTOMERS, CUST_ID == idNum))
    // return;
    SQL * Select(dlg.ctrls).From(CUSTOMERS).Where(CUST_ID == idNum);
    if(!dlg.ctrls.Fetch(SQL))
        return;
    if(dlg.Run() != IDOK)
        return;
    SQL * dlg.ctrls.Update(CUSTOMERS).Where(CUST_ID == idNum);
    CustArray.ReQuery();
    CustArray.FindSetCursor(idNum);
}

void CustomersWindow::AddNewCustomer()
{
    AddCustomer dlg;
    dlg.Title("New Customer");
    if(dlg.Execute() != IDOK)
        return;
    SQL * dlg.ctrls.Insert(CUSTOMERS);
    int id = SQL.GetInsertedId();
    CustArray.ReQuery();
    CustArray.FindSetCursor(id);
}

/*
void CustomersWindow::btnUpdateCustomerClick()
{
    PromptOK(__func__);
}

void CustomersWindow::btnSearchCustomerClick()
{
    PromptOK(__func__);
}
*/

#customers.h

```

```

#ifndef _Invoices_customers_h_
#define _Invoices_customers_h_
#include <Invoices/UI/sqlincludes.h>
#include "../configs.h"

struct CustomersWindow : WithCustomersWindowLayout<TopWindow> {
    Configs myConfig;
    // EditString name;

public:
    CustomersWindow();
    virtual void Paint(Draw& w) {
        w.DrawRect(GetSize(), Color(204, 255, 255)); // <= enter your background color here
    }
    void AddNewCustomer();
    // void btnUpdateCustomerClick();
    // void btnSearchCustomerClick();
    void EditRow();

    // void FakeStub();
};

#endif
#Invoices.Lay
LAYOUT(InvoicesLayout, 460, 116)
END_LAYOUT

LAYOUT(TaxWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(TaxByCustomerWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(IncomeByCustomerWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(ProfitLossWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(SelectDBWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(ShowDBWindowLayout, 400, 200)
END_LAYOUT

LAYOUT(SetCompanyWindowLayout, 400, 200)
END_LAYOUT

```

```
LAYOUT(SelectOutputDirWindowLayout, 400, 200)
END_LAYOUT
```

```
LAYOUT(CustomersWindowLayout, 664, 460)
  ITEM(Button, btnNewCustomer, SetLabel(t_("Add New
Customer"))).SetFrame(ThinInsetFrame()).LeftPosZ(272, 120).TopPosZ(436, 15))
  ITEM(SqlArray, CustArray, AutoHideSb(true).LeftPosZ(4, 656).TopPosZ(4, 420))
END_LAYOUT
```

```
LAYOUT(CustomerAddLayout, 436, 368)
  ITEM(Label, dv___0, SetLabel(t_("Name"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).VSizePosZ(16, 332))
  ITEM(Label, dv___1, SetLabel(t_("Email"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(56, 19))
  ITEM(Label, dv___2, SetLabel(t_("Contact"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(96, 19))
  ITEM(Label, dv___3, SetLabel(t_("Address"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(136, 19))
  ITEM(Label, dv___4, SetLabel(t_("City"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(172, 19))
  ITEM(Label, dv___5, SetLabel(t_("State"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(212, 19))
  ITEM(Label, dv___6, SetLabel(t_("Zip"))).SetFont(SansSerifZ(16)).LeftPosZ(36,
100).TopPosZ(248, 19))
  ITEM(Option, chkTaxable, SetLabel(t_("Taxable?"))).SetFont(SansSerifZ(16)).LeftPosZ(152,
108).TopPosZ(300, 16))
  ITEM(EditString, txtCustName, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
256).TopPosZ(16, 19))
  ITEM(EditString, txtCustEmail, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
256).TopPosZ(56, 19))
  ITEM(EditString, txtCustPhone, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
256).TopPosZ(96, 19))
  ITEM(EditString, txtCustAddress,
SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152, 256).TopPosZ(136, 19))
  ITEM(EditString, txtCustCity, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
256).TopPosZ(172, 19))
  ITEM(EditString, txtCustState, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
256).TopPosZ(212, 19))
  ITEM(EditString, txtCustZip,
MaxChars(10).SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(152,
64).TopPosZ(248, 19))
  ITEM(Button, ok, SetLabel(t_("OK")).LeftPosZ(68, 64).TopPosZ(332, 24))
  ITEM(Button, cancel, SetLabel(t_("Cancel")).LeftPosZ(292, 64).TopPosZ(332, 24))
END_LAYOUT
```

```
LAYOUT(ProductsWindowLayout, 960, 464)
  ITEM(SqlArray, ProductArray, SetFrame(ThinInsetFrame()).LeftPosZ(384, 572).TopPosZ(20,
```

```

440))
ITEM(Label, dv___1, SetLabel(t_("Product ID")).SetFont(SansSerifZ(16)).LeftPosZ(8,
88).TopPosZ(28, 32))
ITEM(Label, dv___2, SetLabel(t_("Date Purchased")).SetFont(SansSerifZ(16)).LeftPosZ(8,
132).TopPosZ(116, 32))
ITEM(Label, dv___3, SetLabel(t_("Description")).SetFont(SansSerifZ(16)).LeftPosZ(8,
88).TopPosZ(160, 32))
ITEM(Label, dv___4, SetLabel(t_("Cost")).SetFont(SansSerifZ(16)).LeftPosZ(8,
88).TopPosZ(304, 32))
ITEM(Label, dv___5, SetLabel(t_("Invoice")).SetFont(SansSerifZ(16)).LeftPosZ(188,
72).TopPosZ(28, 32))
ITEM(EditIntSpin, txtInvoiceNo, Min(-1).NotNull(true).SetFrame(FieldFrame()).LeftPosZ(268,
80).TopPosZ(32, 19))
ITEM(EditString, txtProductName,
SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(104, 248).TopPosZ(76, 19))
ITEM(EditString, txtProductCost,
SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(108, 244).TopPosZ(312, 19))
ITEM(Label, dv___9, SetLabel(t_("Name")).SetFont(SansSerifZ(16)).LeftPosZ(8,
88).TopPosZ(72, 32))
ITEM(EditInt, txtProductID,
Min(1).NotNull(true).SetEditable(false).SetFrame(FieldFrame()).LeftPosZ(104, 64).TopPosZ(32,
19))
ITEM(DropDate, dtpDatePurchased, SetFrame(FieldFrame()).LeftPosZ(152,
104).VSizePosZ(120, 325))
ITEM(Button, btnUpdateProduct,
SetLabel(t_("Update")).SetFont(SansSerifZ(16)).SetFrame(ButtonFrame()).LeftPosZ(108,
84).TopPosZ(352, 28))
ITEM(Button, btnAddProduct,
SetLabel(t_("Add")).SetFont(SansSerifZ(16)).SetFrame(ButtonFrame()).LeftPosZ(16,
84).TopPosZ(352, 28))
ITEM(Button, btnShowAllProduct, SetLabel(t_("Show
All")).SetFont(SansSerifZ(16)).SetFrame(ButtonFrame()).LeftPosZ(16, 84).TopPosZ(408, 28))
ITEM(Button, btnProductRange,
SetLabel(t_("Show")).SetFont(SansSerifZ(16)).SetFrame(ButtonFrame()).LeftPosZ(108,
84).TopPosZ(408, 28))
ITEM(DropDate, dtpDateStart, SetFrame(FieldFrame()).LeftPosZ(248, 104).TopPosZ(384, 19))
ITEM(DropDate, dtpDateEnd, SetFrame(FieldFrame()).LeftPosZ(248, 104).TopPosZ(412, 19))
ITEM(DocEdit, txtDescription, SetFont(SansSerifZ(16)).SetFrame(FieldFrame()).LeftPosZ(108,
244).TopPosZ(168, 128))
END_LAYOUT

```

```

LAYOUT(CreateInvoiceWindowLayout, 1100, 728)
END_LAYOUT

```

```

LAYOUT(ListInvoicesWindowLayout, 1128, 584)
ITEM(SqlArray, ListInvoicesArray, MultiSelect(true).SetFrame(ThinInsetFrame()).RightPosZ(8,
928).VSizePosZ(8, 12))
ITEM(Button, btnPrint, SetLabel(t_("Print Selected")).LeftPosZ(4, 80).TopPosZ(12, 15))

```

```

ITEM(Button, btnApplyPayment, SetLabel(t_("Apply Payment")).LeftPosZ(4, 80).TopPosZ(32,
15))
ITEM(Button, btnEdit, SetLabel(t_("Edit Selected")).LeftPosZ(96, 80).TopPosZ(12, 15))
ITEM(Button, btnVoid, SetLabel(t_("Void Selected")).LeftPosZ(4, 80).TopPosZ(52, 15))
ITEM(Button, btnPaidInFull, SetLabel(t_("Mark PIF")).LeftPosZ(96, 80).TopPosZ(52, 15))
ITEM(Button, btnFixDate, SetLabel(t_("Fix Dates")).LeftPosZ(4, 80).TopPosZ(72, 15))
ITEM(Button, btnByPaid, SetLabel(t_("All Paid")).LeftPosZ(8, 80).TopPosZ(112, 15))
ITEM(Button, btnByBalanceDue, SetLabel(t_("Balance Due")).LeftPosZ(100, 76).TopPosZ(112,
15))
ITEM(LabelBox, dv___9, SetLabel(t_("Show:")).LeftPosZ(4, 180).TopPosZ(92, 124))
ITEM(EditDouble, edbPayment, AlignRight(true).LeftPosZ(96, 80).TopPosZ(32, 19))
ITEM(Button, btnByDates, SetLabel(t_("Date Range")).LeftPosZ(8, 80).TopPosZ(196, 15))
ITEM(Button, btnByCustomer, SetLabel(t_("By Customer")).LeftPosZ(52, 80).TopPosZ(132, 15))
ITEM(Button, btnByVoided, SetLabel(t_("All Voided")).LeftPosZ(8, 80).TopPosZ(172, 15))
ITEM(DropDate, ddRange2, LeftPosZ(92, 88).TopPosZ(192, 19))
ITEM(DropDate, ddRange1, LeftPosZ(92, 88).TopPosZ(168, 19))
ITEM(DropDate, ddFixDate, LeftPosZ(96, 88).TopPosZ(72, 19))
ITEM(DropList, dlCustomers, LeftPosZ(8, 172).TopPosZ(148, 19))
END_LAYOUT

```

```

LAYOUT(ListLineItemsWindowLayout, 1044, 540)
ITEM(SqlArray, ListLineItemsArray, SetFrame(ThinInsetFrame()).LeftPosZ(188, 848).TopPosZ(4,
528))
ITEM(Button, btnFake, LeftPosZ(168, 56).TopPosZ(632, 15))
END_LAYOUT

```

#invoices.sch

```

TABLE_ (CUSTOMERS)
INT_ (CUST_ID) NOT_NULL PRIMARY_KEY AUTO_INCREMENT
STRING_ (CUSTNAME, 150) NOT_NULL
STRING_ (EMAIL, 150)
STRING_ (CONTACT, 150)
STRING_ (ADDRESS, 150)
STRING_ (CITY, 50)
STRING_ (STATE, 20)
STRING_ (ZIP, 10)
INT_ (TAXABLE) NOT_NULL
END_TABLE

```

```

TABLE_ (INVOICES)
INT_ (INVOICE_ID) NOT_NULL PRIMARY_KEY AUTO_INCREMENT
INT_ (INVOICENUMBER) NOT_NULL
INT_ (CUSTOMERID)
INT_ (TRANSACTIONDATE)
STRING_ (TERMS, 20) NOT_NULL
DOUBLE_ (NONTAXABLESUB)
DOUBLE_ (TAXABLESUB)
DOUBLE_ (TAX)

```

```
DOUBLE_ (GRANDTOTAL)
DOUBLE_ (AMTPAID)
INT_ (DATEPAID)
INT_ (STATUS)
END_TABLE
```

```
TABLE_ (PRODUCTS)
INT_ (PROD_ID) NOT_NULL PRIMARY_KEY AUTO_INCREMENT
STRING_ (PRODNAME, 150) NOT_NULL
STRING_ (PRODDescription, 200)
INT_ (DATEPURCHASED)
DOUBLE_ (COST) NOT_NULL
INT_ (INVOICEID) NOT_NULL
END_TABLE
```

```
TABLE_ (LINEITEMS)
INT_ (LINEITEM_ID) NOT_NULL PRIMARY_KEY AUTO_INCREMENT
STRING_ (PRODUCTNAME, 200) NOT_NULL
STRING_ (DESCRIPTION, 200)
DOUBLE_ (PRICE) NOT_NULL
DOUBLE_ (QTY) NOT_NULL
DOUBLE_ (TOTAL) NOT_NULL
INT64_ (INVOICEIDNUMBER) NOT_NULL
INT_ (ISTAXABLE) NOT_NULL
END_TABLE
```

The remaining support files are in the attached zip... Now I should make much more rapid progress until I get to the printing/reporting stage...

Thanks again for all the help :)
Jim

File Attachments

1) [support_files.zip](#), downloaded 258 times
