Subject: How to use multiple schemas and databases?
Posted by jarchalex on Sun, 06 Nov 2011 16:50:53 GMT
View Forum Message <> Reply to Message

Hello!

I'm just studying U++, and cannot figure out how to use multiple databases at once (and possibly multiple schemas). I want application to use several dbs with static info and one to actually write project data.

Probably there is something in the way i should include files to define All_Tables() correctly. But how?..

Please help!

N.B.Attached package is for studying purposes only.

Code that does not work:

```
#include <CtrlLib/CtrlLib.h>
#include <GridCtrl/GridCtrl.h>
#include <Painter/Painter.h>
#include <SqlCtrl/SqlCtrl.h>
#include <plugin/sqlite3/Sqlite3.h>
using namespace Upp;

#define MODEL <studyDraw_01/db.sch>
#define SCHEMADIALECT <plugin/sqlite3/Sqlite3Schema.h>
#include "Sql/sch_header.h"
#include "Sql/sch_source.h"
#include "Sql/sch_schema.h"
/*
#define MODEL <studyDraw_01/db_info.sch>
#define SCHEMADIALECT <plugin/sqlite3/Sqlite3Schema.h>
#include "Sql/sch_header.h"
#include "Sql/sch_source.h"
#include "Sql/sch_schema.h"
*/
#define MODEL <studyDraw_01/db_info.sch>
//#include "Sql/sch_schema.h"


#define LAYOUTFILE <studyDraw_01/gui.lay>
#include <CtrlCore/lay.h>

struct MyDraw: Ctrl {
 virtual void Paint(Draw& w) {
  Size sz = GetSize();
```

```
  ImageBuffer ib(sz);
  BufferPainter sw(ib);
  DoPaint(sw);
  w.DrawImage(0, 0, ib);
 }
 void DoPaint(Painter& sw) {
  Size sz = GetSize();
  sw.DrawRect(0, 0, sz.cx, sz.cy, White());
 }
};
struct MyWindow: TopWindow {
 MenuBar menu;
 StatusBar status;
 SqlArray table1;
 SqlArray tab_settings;
 GridCtrl table2;
 GridCtrl table3;
 TabCtrl tabs;
 Splitter spl, set;
 EditString name;
 EditInt val;
 EditDouble valD;
 EditString parameter;
 EditString value;
 MyDraw draw;
 Sqlite3Session sqlite3, sqlite3_2;

 void InitDB() {
  if(!sqlite3.Open(ConfigFile("settings.conf"))) {
   Exclamation("Can't create or open database file\n");
   return;
  }
  SQL = sqlite3;
  SqlSchema sch(SQLITE3);
  sqlite3.SetTrace();
  All_Tables(sch);
  if(sch.ScriptChanged(SqlSchema::UPGRADE))
   Sqlite3PerformScript(sch.Upgrade());
  if(sch.ScriptChanged(SqlSchema::ATTRIBUTES))
   Sqlite3PerformScript(sch.Attributes());
  if(sch.ScriptChanged(SqlSchema::CONFIG)) {
   Sqlite3PerformScript(sch.ConfigDrop());
   Sqlite3PerformScript(sch.Config());
  }
  sqlite3.SetTrace();

  if(!sqlite3_2.Open(ConfigFile("settings_2.conf"))) {
   Exclamation("Can't create or open database file\n");
```

```
  return;
 }
 //SQL = sqlite3_2;

 SqlSchema sch_2(SQLITE3);
 sqlite3_2.SetTrace();
 All_Tables(sch_2);
 if(sch_2.ScriptChanged(SqlSchema::UPGRADE))
  Sqlite3PerformScript(sch_2.Upgrade());
 if(sch_2.ScriptChanged(SqlSchema::ATTRIBUTES))
  Sqlite3PerformScript(sch_2.Attributes());
 if(sch_2.ScriptChanged(SqlSchema::CONFIG)) {
  Sqlite3PerformScript(sch_2.ConfigDrop());
  Sqlite3PerformScript(sch_2.Config());
 }
 sqlite3.SetTrace();
}
void Exit() {
 if(PromptOKCancel("Exit?"))
  Break();
}

void SubMenu(Bar& bar) {
 bar.Add("Exit", THISBACK(Exit))
  .Help("Exit application");
}

void MainMenu(Bar& bar) {
 bar.Add("Menu", THISBACK(SubMenu));
}

typedef MyWindow CLASSNAME;

MyWindow() {
 InitDB();

 AddFrame(menu);
 AddFrame(status);


 table1.SetSession(sqlite3_2);
 table1.SetTable(FLOWCONVERSIONUNITS);
 table1.AddKey(ID);
 table1.AddColumn(UNIT, t_("Units")).Edit(parameter);
 table1.AddColumn(FACTOR, t_("Conversion factor")).Edit(value);
 table1.Appending().Removing();
 table1.SetOrderBy(ID, UNIT);
```

```
  table2.AddIndex();
  table2.AddColumn(0, t_("One"));
  table2.AddColumn(t_("Two"));
  table2.AddColumn(t_("Three"));
  table2.Appending().Removing().Editing().Accepting().Canceling();
  table2.RejectNullRow();
  table2.SetToolBar();
  table3.AddColumn(0, t_("One"));
  table3.SetToolBar();

  tabs.Add(table1.SizePos(), "table1");

  tabs.Add(table2.SizePos(), "table2");
  tabs.Add(table3.SizePos(), "table3");
  tabs.Set(0);

  tab_settings.SetSession(sqlite3);
  tab_settings.SetTable(SETTINGS);
  tab_settings.AddKey(ID);
  tab_settings.AddColumn(PARAMETER, t_("Parameter")).Edit(parameter);
  tab_settings.AddColumn(VALUE, t_("Value")).Edit(value);
  tab_settings.Appending().Removing();
  tab_settings.SetOrderBy(ID, PARAMETER);

  spl.Vert();
  spl.Add(tab_settings);
  spl.Add(draw);
  spl.Add(tabs);
  spl.SetPos(0,0);
  Add(spl);
  menu.Set(THISBACK(MainMenu));
  menu.WhenHelp = status;
  tab_settings.Query();
 }
};

GUI_APP_MAIN
{
 MyWindow w;
 w.Sizeable().MinimizeBox().MaximizeBox();
 w.SetRect(0, 0, 600, 500);
 w.Run();
}
```

File Attachments

1) studyDraw_01.zip, downloaded 299 times