
Subject: Re: Very Simple Report Generator (Use QTF format)

Posted by [forlano](#) on Sun, 14 Mar 2010 15:55:46 GMT

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

sergeynikitin wrote on Tue, 22 December 2009 23:14I make my 3 cents in the common fund decisions.

For my needs, I made a simple report editor which allows users to easily edit the reporting forms.

Allows:

- Substitute the arbitrary variables and functions (defined at the stage of application programming);
- Generate reports on the database (well, or for any tabular data);
- Insert images, and graphics functions (including data from the database);

Hello,

I was trying to put the Sergey's test case package in the form of class. But I'm having problem with the callbacks in the constructor. I want they become member functions of the class but I do not know how to do it. At the moment the compiler complains. Perhaps the fix is very easy. Here is my experiement. I called the class MakeBadge:

Thank you,
Luigi

```
#include <RepGen/RepGen.h>
using namespace Upp;
```

```
class MakeBadge {
    Vector<String> STRINGS;
    int STRINGS_COUNT;
    int STRINGS_I;
    int TOTAL;

    RepGen rep;
    void ClentCallbackReportVar();
    void ClentCallbackReportFinish();
    void ClentCallbackCalcStart();
    void ClentCallbackCalcBody();
    void ClentCallbackCalcFinish();
    MakeBadge();
}
```

```
//Client Callbacks
void MakeBadge::ClentCallbackReportVar() {
```

```

};

void MakeBadge::ClentCallbackReportFinish() {
    rep.SubstVar("##TOTAL",Format("%`",TOTAL));
};

void MakeBadge::ClentCallbackCalcStart() {
    STRINGS_COUNT = STRINGS.GetCount();
    STRINGS_I = 0;
    TOTAL = 0;
};
void MakeBadge::ClentCallbackCalcBody() {
    //LOG("ClentCallbackCalcBody");
    Vector<String> V = Split(STRINGS[STRINGS_I],',',false);
    //DUMPC(V);

    rep.SubstVarInLoopBody("##PROJECTN",TrimBoth(V.At(0)));
    rep.SubstVarInLoopBody("##PROJECTN",TrimBoth(V.At(0)));
    rep.SubstVarInLoopBody("##PROJECT`_NAME",TrimBoth(V.At(1)));
    rep.SubstVarInLoopBody("##DETAILID",TrimBoth(V.At(2)));
    rep.SubstVarInLoopBody("##COMPANY",TrimBoth(V.At(3)));
    rep.SubstVarInLoopBody("##WAREH`_COMP",TrimBoth(V.At(4)));
    rep.SubstVarInLoopBody("##QUANT",TrimBoth(V.At(5)));
    rep.SubstVarInLoopBody("##SIT",TrimBoth(V.At(6)));
    String pictaddr;

    if (V.GetCount())>=8 && !TrimBoth(V.At(7)).IsEmpty()) {
        pictaddr = TrimBoth(V.At(7));
        if (!pictaddr.IsEmpty()) {
#ifdef PLATFORM_X11
            pictaddr = GetHomeDirectory()+"/MyApps/RepGenTest/"+pictaddr;
            DUMP(pictaddr);
#endif
            Image im = StreamRaster::LoadFileAny(pictaddr);
            rep.PlacelImageInLoopBody( "##IMAGE", im, Size(1100,700) );
            // rep.PlacelImageInLoopBody( "##IMAGE", im );
        }
    }
    else {
        rep.SubstVarInLoopBody("##IMAGE","");
    }
    TOTAL += 1;
    if(++STRINGS_I==STRINGS_COUNT) rep.LoopDone();
};

```

```
void MakeBadge::ClentCallbackCalcFinish() {
    //LOG("ClentCallbackCalcFinish");

};

MakeBadge::MakeBadge()
{
    STRINGS = Split(LoadFile("DATA.csv"), '\n', true);

    rep.RepGenReportVar      = callback(ClentCallbackReportVar);
    rep.RepGenReportFinish   = callback(ClentCallbackReportFinish);
    rep.RepGenCalculateStart = callback(ClentCallbackCalcStart);
    rep.RepGenCalculateBody  = callback(ClentCallbackCalcBody);
    rep.RepGenCalculateFinish = callback(ClentCallbackCalcFinish);

    SetLanguage(GetSystemLNG() & 0xfffff);

    rep.SetTemplate(LoadFile("REPORT_TEMPLATE.QTF"));

    rep.Perform();
}
```
