Subject: Re: howto best Ctrl Refresh handling w/ MT & very frequent refreshes
Posted by Sender Ghost on Fri, 28 May 2010 10:53:22 GMT
View Forum Message <> Reply to Message

Hello, Jan.

Another sample:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

template <class T>
class CachedCtrl : public T {
private:
 Value value;
 bool change;
public:
 CachedCtrl() : change(false) { }
 Callback operator<<=(Callback action) { return T::operator<<=(action); }
 void operator<<=(const Value& data) { change = true; value = data; }
 void SetData(const Value& data) { change = true; value = data; }
 Value GetData() const { return value; }
 Value operator~() const { return value; }
 void Apply() { if (change) T::SetData(value); }
 bool IsChanged() { return change; }
};

const int NUM_CTRLS = 10,
 REFRESH_RATE = 50; // ms

class App : public TopWindow {
private:
 bool doing;
public:
 typedef App CLASSNAME;
 App();
 ~App();

 Thread work;
 typedef CachedCtrl<EditInt> CEditInt;
 Array<CEditInt> ctrls;

 void ChangeData();
 void UpdateData();
 void LeftDown(Point p, dword keyflags);
 void RightDown(Point p, dword keyflags);
};
```

```
App::App() : doing(false)
{
 Title("CachedCtrl test application");
 CenterScreen().Sizeable().MinimizeBox().MaximizeBox();
 SetRect(Size(640, 480));

 for (int i = 0; i < NUM_CTRLS; ++i)
 {
  ctrls.Add().HSizePosZ(4, 4).TopPosZ(4 + i*(19 + 4), 19).SetData(i + 1);
  ctrls[i].Apply();
  Add(ctrls[i]);
 }
}

App::~App()
{
 Thread::ShutdownThreads();
 work.Wait();
}

void App::ChangeData()
{
 if (!doing) doing = true;
 else return;

 const PaintRect curRect = GetBackground();
 Background(PaintRect(ColorDisplay(), SColorPaper()));
 work.Run(THISBACK(UpdateData));

 while (doing)
 {
  Sleep(1);
  GuiLock __;
  if (Thread::IsShutdownThreads()) break;

  for (int i = 0; i < NUM_CTRLS; ++i)
  {
   ctrls[i].SetData(int(~ctrls[i]) % (NUM_CTRLS * 100) + 1);
  }
 }

 Background(curRect);
 doing = false;
}

void App::UpdateData()
{
```

```
 while (doing)
 {
  Sleep(REFRESH_RATE);
  GuiLock __;
  if (Thread::IsShutdownThreads()) break;

  for (int i = 0; i < NUM_CTRLS; ++i)
  {
   ctrls[i].Apply();
  }
 }
}

void App::LeftDown(Point p, dword keyflags)
{
 doing = !doing;
}

void App::RightDown(Point p, dword keyflags)
{
 work.Run(THISBACK(ChangeData));
}

GUI_APP_MAIN
{
 Ctrl::GlobalBackPaint();

 App app;
 app.Run();
}
```

But may be here needed manual refresh mode for all Ctrls instead of automatic (fullrefresh). No need to store extra data. Also IsChanged() equals IsModified() in last case.

For example, in .Net Framework:

```
// Stop refreshing
ctrl.BeginUpdate();
// Change data
// ...
// Start refreshing again
ctrl.EndUpdate();
```