
Subject: Re: Problem with Drag&Drop in ArrayCtrl
Posted by [koldo](#) on Thu, 13 Aug 2009 11:42:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello all

I have prepared a MT version using PostCallback that works very well.

Main change is in function LoadImageThread that substitute the timer function.

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

```
using namespace Upp;
```

```
Array <Image> imgLst;
```

```
struct TestDisplay : Display {  
    virtual void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style)  
    const  
    {  
        w.DrawRect(r, paper);  
        int id = q.Is<int>() ? (int)q : -1;  
        if (id >= 0 && id < imgLst.GetCount()) {  
            Rect rect(r.left, r.top, r.left+70, r.top+70);  
            if (imgLst[id])  
                w.DrawImage(rect, imgLst[id]);  
            else  
                Display::Paint(w, r, "Image not loaded", ink, paper, style);  
        } else  
            Display::Paint(w, r, "Image not available", ink, paper, style);  
    }  
};
```

```
struct App : TopWindow {  
    ArrayCtrl a, b;  
    Splitter s;  
    Thread threadPainter;  
  
    void RefreshThumb(int row)  
    {  
        a.RefreshRow(row);  
    }  
    void DnD(PasteClip& d)  
    {  
        if(AcceptText(d)) {  
            a.Add(GetString(d), GetString(d));  
            a.SetFocus();  
        }  
    }  
};
```

```

}

void DnDInsert(int line, PasteClip& d)
{
if(AcceptInternal<ArrayCtrl>(d, "array")) {
a.InsertDrop(line, d);
a.SetFocus();
}
if(AcceptText(d)) {
a.Insert(line);
a.Set(line, 0, GetString(d));
a.SetCursor(line);
a.SetFocus();
}
}

void DnDInsertB(int line, PasteClip& d)
{
if(AcceptInternal<ArrayCtrl>(d, "array")) {
const ArrayCtrl& src = GetInternal<ArrayCtrl>(d);
bool self = &src == &b;
Vector< Vector<Value> > data;
for(int i = 0; i < src.GetCount(); i++)
if(src.IsSel(i)) {
Value v = src.Get(i, 0);
data.Add().Add(IsNumber(v) ? FormatIntRoman((int)src.Get(i, 0)) : String(v));
}
b.InsertDrop(line, data, d, self);
b.SetFocus();
}
}

void Drag()
{
if(a.DoDragAndDrop(InternalClip(a, "array")) == DND_MOVE)
a.RemoveSelection();
}

void DragB()
{
if(b.DoDragAndDrop(InternalClip(b, "array"), b.GetDragSample()) == DND_MOVE)
b.RemoveSelection();
}

typedef App CLASSNAME;

App();
};

```

```

void LoadImageThread(App *gui, Array <Image> *imgs)
{
for (int i = 0; i < (*imgs).GetCount(); ++i) {
for (int j = 0; j < gui->a.GetCount(); ++j) {
Value ix = gui->a.Get(j, 0);
if (ix.Is<int>() && i == ix) {
if (!(*imgs)[ix]) {
(*imgs)[ix] = StreamRaster::LoadFileAny("/mnt/C/demo.jpg");
PostCallback(callback1(gui, &App::RefreshThumb, i));
}
}
}
}
}
}

```

```

App::App()
{
a.AddColumn("You can paste the text here too").SetDisplay(Single<TestDisplay>());
a.MultiSelect();
a.WhenDropInsert = THISBACK(DnDInsert);
a.WhenDrop = THISBACK(DnD);
a.WhenDrag = THISBACK(Drag);
a.SetLineCy(70);

b.AddColumn("Roman numbers");
b.MultiSelect();
b.WhenDropInsert = THISBACK(DnDInsertB);
b.WhenDrag = THISBACK(DragB);

Add(s.Horz(a, b));
for(int i = 0; i < 25; i++) {
a.Add(i);
imgLst.Add();
b.Add(FormatIntRoman(i, true));
}
Sizeable();

threadPainter.Run(callback2(LoadImageThread, this, &imgLst));
}

GUI_APP_MAIN
{
App().Run();
}

```

This is the output

An annoying problem is that RefreshRow(row) works well in general, but if I just drag an image and does not click or roll the mouse, the image refreshing is stopped like in the image. This behavior happens both in Linux and in Windows. If I click or roll the mouse all the loaded images are refreshed immediately.

Just changing RefreshRow(row) with UpdateRefresh() solves the problem.

Best regards
koldo

File Attachments

1) [dib.PNG](#), downloaded 635 times
