

I have tried to create a testcase:

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

using namespace Upp;

struct TestDisplay : Display {
    virtual void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style)
    const
    {
        Display::Paint(w, r, AsString(q) + " " + AsString(GetTickCount()),
            ink, paper, style);
    }
};

struct App : TopWindow {
    ArrayCtrl a, b;
    Splitter s;
    TimeCallback tb;

    void Timer()
    {
        a.UpdateRefresh();
    }

    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() {
    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);

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

    Add(s.Horz(a, b));
    for(int i = 0; i < 200; i++) {

```

```
a.Add(i);
b.Add(FormatIntRoman(i, true));
}
Sizeable();

tb.Set(-200, THISBACK(Timer));
}
};

GUI_APP_MAIN
{
    App().Run();
}
```

I believe this is as close as possible to your app - and it seems to work quite fine both in linux and win32.

Mirek
