
Subject: Re: Drag and Drop between instances [FEATURE REQUEST]

Posted by [nixnixnix](#) on Wed, 12 Jan 2011 01:37:52 GMT

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

Thanks Mirek,

That all works easy enough. However, an extra complication is that sometimes I DnD between instances but most of the time within the same instance. My code looks like this:

```
void LayerTree::DropInsert(int parent, int ii, PasteClip& d)
{
    AdjustAction(parent, d);
    if(AcceptInternal<LayerTree>(d, "mytreedrag")) {
        const TreeCtrl &src = GetInternal<LayerTree>(d);
        Vector<int> sel = src.GetSel();
        SaveStateToLayers(); // we need to rebuild the tree
        for(int i=0;i<sel.GetCount();i++)
        {
            Drop(parent,sel[i],ii);
        }
        m_ptr->SetTree();
        SetFocus();
        return;
    }
    else if(d.Accept("externalInstance"))
    {
        m_ptr->SetStatus("dropped");

        stringstream ss(d.Get());

        Layer* pLayer = Layer::Load(ss);

        if(pLayer==NULL)
            return;

        m_ptr->AddLayer(pLayer);
    }

    // if(AcceptText(d))
    // {
    //     SetCursor(Insert(parent, ii, Image(), GetString(d)));
    //     SetFocus();
    //     return;
    // }
}
```

```

void LayerTree::Drag()
{
    if(m_ptr->AreWeBusy())
        return;

    if(DoDragAndDrop(InternalClip(*this, "mytreedrag"),
                     this->GetDragSample()) == DND_MOVE)
    {
        RemoveSelection();
    }
}

void LayerTree::DragLeave()
{
    int id;

    id = GetCursor();
    if(id<=0) // if id==0 then its the root node which is not a layer
        return;

    // copy to clipboard
    TreeCtrl::Node node = GetNode(id);
    LayerOption* ptr = (LayerOption*)~node.ctrl;
    Layer* pLayer = ptr->GetLayer();

    stringstream ss;

    ss.SetStoring();

    ss.Put("layer");
    int type = int(pLayer->GetType());
    ss.Put32(type);

    pLayer->Serialize(ss);

    String sLayer(ss);

    String text = pLayer->GetName();

    Size isz = GetTextSize(text.ToWString(), StdFont());

    ImageDraw iw(isz);

    iw.DrawRect(isz, White);

    iw.DrawText(0, 0, text);

    VectorMap<String, ClipData> clip;

```

```
clip.Add("externalInstance", sLayer);  
  
if(DoDragAndDrop(clip, iw) == DND_MOVE)  
{  
  
}  
  
TreeCtrl::DragLeave();  
  
}
```

My questions are: do I need to treat my internal drag and drops the same way as my external ones? Is there an external and an internal DnD working simultaneously? Is there a way to know which instance a DnD is coming from?

It all seems to work fine so long as my cursor doesn't leave and return to the same instance.

Cheers,

Nick