
Subject: F2 tree editor...

Posted by [fudadmin](#) on Tue, 11 Apr 2006 13:56:20 GMT

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

```
/*
 * to make this work you need Dirty(id) instead of RefreshItem(id) in TreeCtrl.cpp...
 * void TreeCtrl::SetNode(int id, const TreeCtrl::Node& n)
 * triple click removes all selection - handy... :)
 * in some cases, when cursor is moved, after Enter, Dirty(id) doesn't work properly, I guess...
 * todo: when lost focus - some functionality...
 * todo: font sizes
 * NodeEditor should go inside TreeEdit or at least some methods from it?
 */
#include <CtrlLib/CtrlLib.h>

class NodeEditor : public EditString {
protected:
    int mcursor;
public:

    Size GetMinSize() const;
    Size GetMinFitSize();

    int GetCursor()    {return cursor;}
    int GetMCursor()   {return mcursor;}
    void SetMCursor()  {mcursor=cursor;}

    typedef NodeEditor CLASSNAME;

    NodeEditor() {}
    ~NodeEditor(){}
};

class TreeEdit : public TreeCtrl {

private:
    NodeEditor editor;
    Value oldvalue;
    int editid;
    TreeCtrl::Node editnode;

public:

    virtual bool HotKey(dword key);

    Value GetOldValue()  {return oldvalue;}
};
```

```

void RemoveEditAndUpdate(const Value& value);
void AddEditToNode();
void NodeAutoSize();

```

```

typedef TreeEdit CLASSNAME;

```

```

TreeEdit();
~TreeEdit(){;}
};

```

```

//////////=====
Size NodeEditor::GetMinFitSize() //not elegant but because of that const!
{
    Size sz = StdDisplay().GetStdSize(GetData());
    sz += Size(2 * 4, 6); //adding some margins...
    return sz;
}

```

```

Size NodeEditor::GetMinSize() const //virtual for TreeCtrl::Node.ctrl->...
{
    Size sz = StdDisplay().GetStdSize(GetData());
    sz += Size(2 * 4, 6); //adding some margins...
    return sz;
}

```

```

//////////=====
void TreeEdit::NodeAutoSize()
{
    Size sz = editor.GetMinFitSize();
    editor.SetMCursor();
    SetNode(editid, editnode.SetSize(sz));
    NoCursor(); //some cursor tricks to prevent all text selection...
    editor.Move(editor.GetMCursor(),false);
    NoCursor(false);
    SetCursor(editid); //it looks like after NoCursor it goes to root...
}

```

```

bool TreeEdit::HotKey(dword key)
{
    switch(key)
    {
    case K_F2:
        AddEditToNode();
        // break;
        return true;
    }
}

```

```

case K_ESCAPE:
    RemoveEditAndUpdate( GetOldValue() );
    // break;
    return true;
case K_ENTER:
    RemoveEditAndUpdate( editor.GetData() ); //update despite focus was lost - what I want
sometimes! (add set...)
    // break;
    return true;
}
return Ctrl::HotKey(key);
// return false;
}

```

```

void TreeEdit::RemoveEditAndUpdate(const Value& value)
{
    if (editid>=0) {
        editor.SetRect(0,0,0,0); //need to kill it better?

        editnode.SetSize(Null); //to make auto back from Value...
        SetNode(editid, editnode.Set(value)); //ok
        SetFocus(); //need this for cursor restore
        editid=-1; //might change for undo history
    }
}

```

```

void TreeEdit::AddEditToNode()
{
    if (editid>=0) RemoveEditAndUpdate(GetOldValue()); //need more clever way? ctrl?

    editid = GetCursor(); //ok
    editnode = GetNode(editid); //ok

    oldvalue = Get(editid); //store in private for restore
    editor.SetData(oldvalue); //ok

    SetNode(editid, editnode.SetCtrl(editor)); //ok
    editor.SetSelection(0); //to select all in editor...
    // editor.Move(editor.GetLength(),true); //?or if you don't want selection...
    editor.Action(); //to enable left-right keys at once...
}

```

```

TreeEdit::TreeEdit()
{
    editor.WhenAction = THISBACK(NodeAutoSize);
}

```

```

WhenLeftDouble = THISBACK(AddEditToNode);
editid = -1;
oldvalue = NULL;
}

```

```

//////////=====

```

```

class TreeEditor : public TopWindow {
    TreeEdit tree;
    ArrayCtrl arr;
    Splitter splitter;
public:

```

```

    typedef TreeEditor CLASSNAME;

```

```

    TreeEditor();
};

```

```

TreeEditor::TreeEditor()

```

```

{
    splitter.Add(tree);
    splitter.Add(arr);
    splitter.Horz().SetPos(3000,0);

```

```

    Add(splitter.HSizePos(10, 10).VSizePos(40, 40));

```

```

    arr.AddColumn("name", "Name", 4);
    arr.AddColumn("size", "Size", 2);
    arr.AddColumn("type", "Type", 2);
    arr.AddColumn("datemod", "Date Modified", 3);
    arr.AddColumn("empty", "", 2);

```

```

    tree.SetRoot(Ctrllmg::Dir(), "just a root...");
    tree.Open(0);

```

```

    tree.Add(0, Ctrllmg::File(), "item 1");
    tree.Add(1, Ctrllmg::File(), "item 10");

```

```

    tree.Add(0, Ctrllmg::File(), "item 2 blala");
    tree.Add(0, Ctrllmg::File(), "item 3");
    tree.Add(0, Ctrllmg::File(), "item 3");
    tree.Add(0, Ctrllmg::File(), "item 3");
    tree.Add(0, Ctrllmg::File(), "item 3");

```

```

}

```

```
GUI_APP_MAIN
```

```
{  
  TreeEditor().Title("Aris F2 Tree Editor :) v1").Sizeable().Zoomable().Run();  
}
```

File Attachments

1) [F2TreeEditor.zip](#), downloaded 1889 times

Subject: Re: F2 tree editor...

Posted by [mirek](#) on Wed, 12 Apr 2006 13:47:05 GMT

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

Ok, there is now EditField::AutoSize(int maxcx)...

Mirek

Subject: Re: F2 tree editor...

Posted by [fudadmin](#) on Wed, 12 Apr 2006 14:42:39 GMT

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

luzr wrote on Wed, 12 April 2006 14:47Ok, there is now EditField::AutoSize(int maxcx)...

Mirek

I wasn't asking for it... .

Thanks a lot, anyway!
