
Subject: Wrap EditString Rect to text

Posted by [iST1](#) on Sat, 05 Oct 2013 09:08:25 GMT

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

I use EditString field as a TreeCtrl node for editing like in MS Word's list and for hiding its boundary I use `edit.SetFrame(NULLFrame());` But all the same around EditString hidden indents exist. How to wrap border to text?

Subject: Re: Wrap EditString Rect to text

Posted by [mirek](#) on Mon, 07 Oct 2013 18:22:49 GMT

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

Not sure I understand the question, all seems OK to me. Can you provide example and/or commented screenshot?

Subject: Re: Wrap EditString Rect to text

Posted by [iST1](#) on Tue, 08 Oct 2013 16:39:57 GMT

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

Mirek, please check this example

```
class EditableTree : public TreeCtrl {
public:
    typedef EditableTree CLASSNAME;

    EditableTree() : selId_(0) {
        editNode_.SetCtrl(edit_);
        edit_.SetFrame(NULLFrame());
        OpenDeep(0, true);

        WhenSel = THISBACK(OnSelect);
    }

    void AddNode() {
        CloseEdit();

        selId_ = Add(0);
        Open(selId_);
        BeginEdit();
    }

    void BeginEdit() {
        editNode_.image = selNode_.image;
        SetNode(selId_, editNode_);
    }
};
```

```

String editval = "NODE" + AsString(selId_);
    edit_.SetData(editval);
    edit_.Show();
    edit_.AutoSize();
edit_.SetFocus();//todo: at first not select because IsOpen is false*.
edit_.SetSelection(editval.GetCount());
}

```

```

void CloseEdit() {
if (selId_ > 0) {
    selNode_.value = ~edit_;
    SetNode(selId_, Node(~edit_));
    edit_.Hide();
}
}

```

```

void OnSelect()
{
    CloseEdit();
    selId_ = GetSel()[0];
    selNode_ = GetNode(selId_);
    BeginEdit();
}

```

```

private:
int selId_;
    EditString edit_;
    Node selNode_;
    Node editNode_;
};

```

```

GUI_APP_MAIN
{
    TopWindow wnd;
    Button btn;
    wnd.Add(btn.SetLabel("Add").LeftPos(10).TopPos(10));

    EditableTree tree;
    wnd.Add(tree.VSizePos(40).HSizePos());

    btn <<= callback(&tree, &EditableTree::AddNode);
    wnd.Run();
}

```

In edit mode (when select) text shifts in horizontal

Subject: Re: Wrap EditString Rect to text
Posted by [mirek](#) on Tue, 08 Oct 2013 18:30:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, that is because internal margin of EditField.

I have now added NoInternalMargin, which removes it. Use

```
edit_.NoInternalMargin().SetFrame(NullFrame());
```

Mirek

Subject: Re: Wrap EditString Rect to text
Posted by [iST1](#) on Fri, 11 Oct 2013 07:57:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

This code illustrate also a setting user defined font size, where some trouble occur with changing initial text's position

```
enum {fntSize = 19};
```

```
struct AppNodeSize : public Display {  
    virtual void Paint(Draw& w,const Rect& r, const Value& q,Color ink,Color paper,dword style)  
    const {  
        w.DrawRect(r, paper);  
        w.DrawText(r.left, r.top, (String)q, StdFont().Height(fntSize), ink);  
    }  
};
```

```
Size GetStdSize(const Value& q) const {  
    //1.5 is a empirical multiplier for increasing row height when fntSize used in (1) and (2);  
    //how to universally set it based on fntSize?  
    return 1.5 * StdDisplay().GetStdSize(q);  
}  
};
```

```
class EditableTree : public TreeCtrl {  
public:  
    typedef EditableTree CLASSNAME;
```

```
    EditableTree() : selld_(0) {  
        NoRoot();
```

```
        editNode_.SetCtrl(edit_);  
        edit_.NoInternalMargin().SetFrame(NullFrame());//crashes when NoInternalMargin  
        edit_.SetFont(StdFont().Height(fntSize));//(1)
```

```

WhenSel = THISBACK(OnSelect);
}

void AddNode() {
    CloseEdit();

    //with some ico in /**/ offset lines more visible
    selNode_ = Node(/*espudImg::secExpand(), */"NODE" + AsString(GetChildCount(0) + 1));
    selNode_.SetDisplay(Single<AppNodeSize>());//(2)

    selId_ = Add(0, selNode_);
    BeginEdit();
}

void OnSelect() {
    CloseEdit();

    selId_ = GetSel()[0];
    selNode_ = GetNode(selId_);

    BeginEdit();
}

private:
int selId_;
    Node selNode_;

    EditString edit_;
    Node editNode_;

void BeginEdit() {
    editNode_.image = selNode_.image;
    SetNode(selId_, editNode_);

    String val = selNode_.value;
    edit_.SetData(val);
    edit_.AutoSize();
    edit_.Show();
    edit_.SetFocus();
    edit_.SetSelection(val.GetCount());
}

void CloseEdit() {
    if (selId_ > 0) {
        selNode_.value = ~edit_;
        SetNode(selId_, selNode_);
        edit_.Hide();
    }
}

```

```
}  
}  
};
```

Subject: Re: Wrap EditString Rect to text
Posted by [iST1](#) on Fri, 11 Oct 2013 09:46:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another interesting task of setting the cursor when editing based on the current mouse position, i.e. like in Word's list. Additionally with selection on hover. This may be a good example future.

Subject: Re: Wrap EditString Rect to text
Posted by [mirek](#) on Wed, 16 Oct 2013 06:35:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

NoInternalMargin fixed, the method definition you seek is

```
Size GetStdSize(const Value& q) const {  
    return GetTextSize((String)q, StdFont().Height(fntSize));  
}
```

However, there are still some minor problems (selected node is vertically shifting); will fix those soon.

Mirek

P.S.: Complete testcase (for my reference):

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

```
using namespace Upp;
```

```
enum {fntSize = 19};
```

```
struct AppNodeSize : public Display {  
    virtual void Paint(Draw& w,const Rect& r, const Value& q,Color ink,Color paper,dword style)  
    const {  
        w.DrawRect(r, paper);  
        w.DrawText(r.left, r.top, (String)q, StdFont().Height(fntSize), ink);  
    }  
}
```

```
Size GetStdSize(const Value& q) const {  
    return GetTextSize((String)q, StdFont().Height(fntSize));  
}
```

```

}
};

class EditableTree : public TreeCtrl {
public:
    typedef EditableTree CLASSNAME;

    EditableTree() : selId_(0) {
        NoRoot();

        editNode_.SetCtrl(edit_);
        edit_.NoInternalMargin().SetFrame(NullFrame());
        edit_.SetFont(StdFont().Height(fntSize));

        WhenSel = THISBACK(OnSelect);
    }

    void AddNode() {
        CloseEdit();

        //with some ico in /**/ offset lines more visible
        selNode_ = Node(/*espudImg::secExpand(), */"NODE" + AsString(GetChildCount(0) + 1));
        selNode_.SetDisplay(Single<AppNodeSize>());//(2)

        selId_ = Add(0, selNode_);
        BeginEdit();
    }

    void OnSelect() {
        CloseEdit();

        selId_ = GetSel()[0];
        selNode_ = GetNode(selId_);

        BeginEdit();
    }

private:
    int selId_;
    Node selNode_;

    EditString edit_;
    Node editNode_;

    void BeginEdit() {
        editNode_.image = selNode_.image;
        SetNode(selId_, editNode_);
    }

```

```
String val = selNode_.value;
    edit_.SetData(val);
    edit_.AutoSize();
edit_.Show();
edit_.SetFocus();
    edit_.SetSelection(val.GetCount());
}
```

```
void CloseEdit() {
if (selId_ > 0) {
    selNode_.value = ~edit_;
    SetNode(selId_, selNode_);
    edit_.Hide();
}
}
};
```

GUI_APP_MAIN

```
{
    TopWindow wnd;
    Button btn;
    wnd.Add(btn.SetLabel("Add").LeftPos(10).TopPos(10));

    EditableTree tree;
    wnd.Add(tree.VSizePos(40).HSizePos());

    btn <<= callback(&tree, &EditableTree::AddNode);
    wnd.Run();
}
```
