
Subject: Horizontal Cursor for CodeEditor/LineEdit
Posted by [abductee23](#) on Fri, 20 Aug 2021 23:17:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Y'All,

tl;dr since i was frustrated by not having this, i made it and i want to share, so here it goes:

cheers
--M

p.s.: yo Mirek any chance this can make it into the actual CodeEditor or LineEdit?

CoolCodeEditor.h

```
#pragma once
#include <CtrlLib/CtrlLib.h>
#include <CtrlCore/CtrlCore.h>
#include "CodeEditor/CodeEditor.h"

using namespace Upp;

class CoolCodeEditor : public CodeEditor{
public:
    virtual bool Key(dword key, int count);
    typedef CoolCodeEditor CLASSNAME;
};
```

CoolCodeEditor.cpp

```
#include "CoolCodeEditor.h"
#include "CodeEditor/CodeEditor.h"

using namespace Upp;

bool CoolCodeEditor::Key(dword key, int count) {
    int line = GetCursorLine();
    int cursor_pos = GetCursor();
    int line_start = GetPos64(line,0);
    String ss = Get(line_start,cursor_pos-line_start);
    dword real_key = key & ~K_SHIFT;
    //real_key = real_key & ~K_KEYUP;
```

```

Rect r = GetRectSelection();
if( r.left != 0x80000000 && r.left == r.right ){
    int cnt = (r.bottom-r.top)+1;

    //Log("rect = %d %d %d %d, key= %d",r.left, r.top, r.right, r.bottom, real_key);
    if( real_key >= 32 && real_key <= 126 ){
        NextUndo();
        WString txt;
        for( int i = 0; i < cnt;i++){
            txt += WString(real_key,1)+WString("\n");
        }
        PasteRectSelection(txt);
        r.left++;
        r.right++;
        SetRectSelection( r );
        Sync();
        return true;
    }
    if( real_key == K_BACKSPACE ){
        Rect r_back = Rect(r);
        r_back.left--;
        SetRectSelection(r_back);
        RemoveSelection();
        r.left--;
        r.right--;
        SetRectSelection(r);
        return true;
    }
    if( real_key == K_DELETE ){
        Rect r_back = Rect(r);
        r_back.right++;
        SetRectSelection(r_back);
        RemoveSelection();
        SetRectSelection(r);
        return true;
    }
    if( key == K_CTRL_V ){
        WString w = ReadClipboardUnicodeText();
        if(w.IsEmpty())
            w = ReadClipboardText().ToWString();
        if(w.IsEmpty())
            w = GetPasteText().ToWString();

        int len = w.GetLength();
        if( w.IsEmpty() ){
            len=0;

```

```

}
WString txt;
for( int i = 0; i < cnt;i++){
    txt += w+WString("\n");
}
PasteRectSelection(txt);
r.left +=len;
r.right+=len;
SetRectSelection( r );

return true;
}

//SetRectSelection
}
if( r.left != 0x80000000 && r.left < r.right && r.top < r.bottom ){
    int cnt = (r.bottom-r.top)+1;
    if( real_key >= 32 && real_key <= 126 ){
        RemoveSelection();
        r.right=r.left;
        WString txt;
        for( int i = 0; i < cnt;i++){
            txt += WString(real_key,1)+WString("\n");
        }
        SetRectSelection( r );
        PasteRectSelection(txt);
        r.left++;
        r.right++;
        SetRectSelection( r );
        return true;
    }
    if( real_key == K_DELETE || real_key == K_BACKSPACE){
        RemoveSelection();
        r.right=r.left;
        SetRectSelection( r );
        return true;
    }
}

if( key == K_CTRL_V ){
    RemoveSelection();
    r.right=r.left;
    SetRectSelection( r );

    WString w = ReadClipboardUnicodeText();
    if(w.IsEmpty())
        w = ReadClipboardText().ToWString();
    if(w.IsEmpty())
        w = GetPasteText().ToWString();
}

```

```
int len = w.GetLength();
if( w.IsEmpty() ){
    len=0;
}
WString txt;
for( int i = 0; i < cnt;i++){
    txt += w+WString("\n");
}
PasteRectSelection(txt);
r.left +=len;
r.right+=len;
SetRectSelection( r );

return true;
}

}
return LineEdit::Key(key,count);
}
```

File Attachments

1) [CoolCodeEditor.gif](#), downloaded 317 times

```
if( k == 0 ) return vec3(0,0,0);
if( k == 1 ) return vec3(1,0,0);
if( k == 2 ) return vec3(0,1,0);
if( k == 3 ) return vec3(0,0,1);
```
