
Subject: Re: simple text_edit

Posted by [forlano](#) on Wed, 31 May 2006 13:40:54 GMT

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

qwerty wrote on Wed, 31 May 2006 14:39 which component to use if I want Ctrl with simple utf-7 viewing/editing, newlines and tabs? I cant find new lines in QTF, & dont seems like '\n'. And how to use it. There I am quite lost

and, any chance to get, what get can, from richtextedit to text file?

If you need a simple text file editor you should use LineEdit. The code below is the reference example UWord modified with LineEdit. Perhaps it can be useful:

Luigi

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

```
FileSel& TextEditorFs()
{
    static FileSel fs;
    return fs;
}
```

```
class TextEditor : public TopWindow {
protected:
    LineEdit editor;
    MenuBar menubar;
    ToolBar toolbar;
    StatusBar statusbar;
    String filename;
```

```
void Load(const String& filename);
void New();
void Open();
void Save();
void SaveAs();
void Print();
void About();
void Destroy();
void SetBar();
void FileBar(Bar& bar);
void AboutMenu(Bar& bar);
void MainMenu(Bar& bar);
void MainBar(Bar& bar);
```

```

public:
    typedef TextEditor CLASSNAME;

    TextEditor();
};

void TextEditor::FileBar(Bar& bar)
{
    bar.Add("New", CtrlImg::new_doc(), THISBACK(New))
        .Key(K_CTRL_N)
        .Help("Open new window");
    bar.Add("Open..", CtrlImg::open(), THISBACK(Open))
        .Key(K_CTRL_O)
        .Help("Open existing document");
    bar.Add(editor.IsModified(), "Save", CtrlImg::save(), THISBACK(Save))
        .Key(K_CTRL_S)
        .Help("Save current document");
    bar.Add("SaveAs", CtrlImg::save_as(), THISBACK(SaveAs))
        .Help("Save current document with a new name");
    bar.ToolGap();
    bar.MenuSeparator();
    bar.Add("Print..", CtrlImg::print(), THISBACK(Print))
        .Key(K_CTRL_P)
        .Help("Print document");
    if(bar.IsMenuBar()) {
        bar.Separator();
        bar.Add("Exit", THISBACK(Destroy));
    }
}

void TextEditor::AboutMenu(Bar& bar)
{
    bar.Add("About..", THISBACK(About));
    bar.Separator();
    bar.Add("What is this ?", callback(PerformDescription))
        .Key(K_SHIFT_F1);
}

void TextEditor::MainMenu(Bar& bar)
{
    bar.Add("File", THISBACK(FileBar));
    bar.Add("Window", callback(WindowsMenu));
    bar.Add("Help", THISBACK(AboutMenu));
}

void TextEditor::New()
{
    new TextEditor;
}

```

```

}

void TextEditor::Load(const String& name)
{
    editor.Set(LoadFile(name));
    editor.ClearModify();
    Title(filename);
}

void TextEditor::Open()
{
    FileSel& fs = TextEditorFs();
    if(fs.ExecuteOpen())
        if(filename.IsEmpty() && !editor.IsModified())
            Load(fs);
        else
            (new TextEditor)->Load(fs);
    else
        statusBar.Temporary("Loading aborted.");
}

void TextEditor::Save()
{
    if(!editor.IsModified()) return;
    if(filename.IsEmpty())
        SaveAs();
    else
        if(SaveFile(filename, editor.Get())) {
            statusBar.Temporary("File " + filename + " was saved.");
            ClearModify();
        }
    else
        Exclamation("Error saving the file: " + filename);
}

void TextEditor::SaveAs()
{
    FileSel& fs = TextEditorFs();
    if(fs.ExecuteSaveAs()) {
        filename = fs;
        Title(filename);
        Save();
    }
}

void TextEditor::Print()
{
    //editor.Print();
}

```

```

}

void TextEditor::About()
{
    PromptOK("[A5 uNotePad]&Using [*^www://upp.sf.net^ Ultimate`+`+] technology.");
}

void TextEditor::Destroy()
{
    if(editor.IsModified()) {
        switch(PromptYesNoCancel("Do you want to save the changes to the document?")) {
            case 1:
                Save();
                break;
            case -1:
                return;
        }
    }
    delete this;
}

void TextEditor::MainBar(Bar& bar)
{
    FileBar(bar);
    bar.Separator();
}

void TextEditor::SetBar()
{
    toolbar.Set(THISBACK(MainBar));
}

TextEditor::TextEditor()
{
    AddFrame(menuubar);
    AddFrame(TopSeparatorFrame());
    AddFrame(toolbar);
    AddFrame(statusbar);
    Add(editor.SizePos());
    menuubar.Set(THISBACK(MainMenu));
    Sizeable().Zoomable();
    WhenClose = THISBACK(Destroy);
    menuubar.WhenHelp = toolbar.WhenHelp = statusbar;
    static int doc;
    Title(Format("Document%d", ++doc));
    Icon(CtrlImg::File());
    editor.ClearModify();
}

```

```

editor <<= THISBACK(SetBar);
SetBar();
OpenMain();
ActiveFocus(editor);
}

void SerializeApp(Stream& s)
{
    int version = 0;
    s / version;
    s % TextEditorFs();
}

GUI_APP_MAIN
{
    SetLanguage(LNG_ENGLISH);

    TextEditorFs().Type("text files", "*.txt")
        .AllFileType()
        .DefaultExt("txt");

    LoadFromFile(callback(SerializeApp));
    new TextEditor;
    Ctrl::EventLoop();
    StoreToFile(callback(SerializeApp));
}

```
