Subject: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by fudadmin on Thu, 30 Mar 2006 13:57:00 GMT
View Forum Message <> Reply to Message

I want to have my customized DialectCodeEditor, someone PythonCodeEditor etc. with connections to I guess different Assists. And, I guess, we all want different right click popup menus.

What would be the best for all interested sides:
1. a separate UppCodeEditor, PythonCodeEditor etc. by sublassing from CodeEditor or
2. expanding the existing interface?
3. other alternatives.

I've started the 1.option succesfully but I don't want to waste my time if later some new grand changes will emerge. Also, I want to know the best technical considerations.

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by mirek on Thu, 30 Mar 2006 20:03:54 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 30 March 2006 08:57I want to have my customized DialectCodeEditor, someone PythonCodeEditor etc. with connections to I guess different Assists. And, I guess, we all want different right click popup menus.

What would be the best for all interested sides:
1. a separate UppCodeEditor, PythonCodeEditor etc. by sublassing from CodeEditor or
2. expanding the existing interface?
3. other alternatives.

I've started the 1.option succesfully but I don't want to waste my time if later some new grand changes will emerge. Also, I want to know the best technical considerations.

Well, the most safe approach would be to base your efforts on LineEdit (CodeEditor is derived from it as well).

Personally, I would most likely used CodeEditor, mosty to get Find&Replace, and reimplemented highlighting section.

Mirek

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by fudadmin on Thu, 30 Mar 2006 20:15:04 GMT
View Forum Message <> Reply to Message

But where to insert the code for popup context menu?
I've made my DialectCodeEditor derived from CodeEditor with adjusted HighlightLine and extra menu items.

something like this:

class DEditor : public CodeEditor {

```
    virtual  void HighlightLine(int line, Vector<LineEdit::Highlight>& hl, int pos);
public:
 void    OnListMenu(Bar& bar);
 void ToolEditInsert(Bar& bar);
 void testInsert();

 typedef DEditor CLASSNAME;

 DEditor();
 ~DEditor() {;}
};
```

It works quite nicely (I rewrote HighlightLine implementation)
Why do you think I should use LineEdit?

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by mirek on Thu, 30 Mar 2006 20:20:20 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 30 March 2006 15:15But where to insert the code for popup context menu?

LineEdit::WhenBar

Quote:
It works quite nicely (I rewrote HighlightLine implementation)
Why do you think I should use LineEdit?

Well, as part of CtrlLib, it has quite fixed interface.

However, in fact, I suggested that just as option. I believe that deriving CodeEditor is a way to go.

Mirek

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by fudadmin on Thu, 30 Mar 2006 20:28:28 GMT

View Forum Message <> Reply to Message

luzr wrote on Thu, 30 March 2006 21:20fudadmin wrote on Thu, 30 March 2006 15:15But where to insert the code for popup context menu?

LineEdit::WhenBar

Mirek

I know that! It works for me. I even made dynamic menus. But I'am asking in which place of the uppsrc/ide files or classes you are going to make connections with existing Assist menu items like "Go to definition/declaration"? Are you going to change LineEdit?!

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by mirek on Thu, 30 Mar 2006 20:32:41 GMT

View Forum Message <> Reply to Message

fudadmin wrote on Thu, 30 March 2006 15:28luzr wrote on Thu, 30 March 2006 21:20fudadmin wrote on Thu, 30 March 2006 15:15But where to insert the code for popup context menu?

LineEdit::WhenBar

Mirek

I know that! It works for me. I even made dynamic menus. But I'am asking in which place of the uppsrc/ide files or classes you are going to make connections with existing Assist menu items like "Go to definition/declaration"? Are you going to change LineEdit?!

Not sure what the question is... If you are asking about how I plan to add assist++ items to CodeEditor (which indeed should happen), then LineEdit::WhenBar is a perfect way how to do that. And it will likely be in the "ide" package, not "CodeEditor"...

Mirek

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by fudadmin on Thu, 30 Mar 2006 20:38:51 GMT

Quote:Not sure what the question is... If you are asking about how I plan to add assist++ items to CodeEditor (which indeed should happen), then LineEdit::WhenBar is a perfect way how to do that. And it will likely be in the "ide" package, not "CodeEditor"...

Could you be more precise? I want to guess your logic of thinking and experiment.

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by mirek on Thu, 30 Mar 2006 20:41:45 GMT

fudadmin wrote on Thu, 30 March 2006 15:38Quote:Not sure what the question is... If you are asking about how I plan to add assist++ items to CodeEditor (which indeed should happen), then LineEdit::WhenBar is a perfect way how to do that. And it will likely be in the "ide" package, not "CodeEditor"...

Could you be more precise? I want to guess your logic of thinking and experiment.

Ide::Ide
{
......
    editor.WhenBar = THISNACK(IdeEditorLocalMenu);
......
}

Is that what you have on mind? (I am not sure).

Mirek

Subject: Re: What is the best: a separate UppCodeEditor or expanding the existing?
Posted by fudadmin on Thu, 30 Mar 2006 22:40:16 GMT

And then in idebar.cpp...
Quote:
void Ide::IdeEditorLocalMenu(Bar& menu)
{   //what items are needed put here?...
    menu.Add(AK_JUMPS, THISBACK(JumpS));

```
menu.Add(AK_SWAPS, THISBACK(SwapS));
menu.Add(AK_JUMPS, THISBACK(JumpS));
menu.Separator();
editor.StdBar(menu);
    //add extra item "customize menu..."
}
```

Thanks! ...Need to add some "if" logic here...

Edit: I mean, it works