Subject: Suggestions for improvements
Posted by aroman on Sun, 20 Nov 2005 19:29:21 GMT
View Forum Message <> Reply to Message

Most of these suggestions are taken from Visual Assist for Visual Studio. It's an amazing add-on, and including most of its functionality into TheIDE would be huge. The visual assist website very clearly documents most of its extensions, and so I've linked to many of them here so you can see what I mean.

Assist++

Show autocomplete on "::" in addition to "." and "->". Especially for calling static class functions and filling out member functions (if for some reason I forgot to alt-c).
Show function prototype above function during autocompletion. See http://www.wholetomato.com/products/features/parameter.html
Show non-exact case matches in completion window. Don't remove matches that don't match the exact case, but just list exact-case matches first and non-exact case matches next.
Insert the top match if the user hits TAB. (Saves having to hit down arrow and enter, which make the hand move far.)
Display the prototype definition line of the current symbol under the cursor at the top of the editing window. Or, as in visual slickedit, display the snippet of code around the definition of the current symbol under the cursor. See http://www.wholetomato.com/products/features/definition.html and similarly http://www.slickedit.com/images/stories/screenshots/vs10_jav a1.jpg
Insert closing })] " when the corresponding opening character is typed. See http://www.wholetomato.com/products/features/insert.html
Show possible header files during #include. See http://www.wholetomato.com/products/features/include.html

Other

Improve "Enclose selected text". Allow using '/' to comment lines with // and '*' to comment lines with /* ... */. Also, if the lines are already commented, then un-comment them instead. See http://www.wholetomato.com/products/features/surround.html
Column indicator - draw faint dashed line at column 80. User-configurable line color and column number. See http://www.wholetomato.com/products/features/column.html
Switch between cpp and h file. In Visual Assist, Alt-O is "open opposite" -- if you're in a .h file, it will open the corresponding C file and vice versa. Very convenient! See http://www.wholetomato.com/products/features/header.html

Some others:
http://www.wholetomato.com/products/features/case.html
http://www.wholetomato.com/products/features/suggestion.html
http://www.wholetomato.com/products/features/move.html

TheIDE is already pretty nice, but with these improvements it could really compete with Visual

Studio and dominate editors under linux, IMO.

- Augusto

Subject: Re: Suggestions for improvements
Posted by mirek on Sun, 20 Nov 2005 21:43:48 GMT
View Forum Message <> Reply to Message

Thanks. Some of them are already in ToDo, some of them can be done easily, some do not expect soon

---

Subject: Re: Suggestions for improvements
Posted by mr_ped on Tue, 22 Nov 2005 21:48:07 GMT
View Forum Message <> Reply to Message

aroman wrote on Sun, 20 November 2005 20:29
Show possible header files during #include.

And after right clicking on
#include "some_h_file.h"
add "Open file: some_h_file.h" into context menu.

And another minor improvement:
Allow to reorganize file tabs with drag and drop method.
(like new firefox 1.5 has it)

Subject: Re: Suggestions for improvements
Posted by mirek on Tue, 22 Nov 2005 22:32:08 GMT
View Forum Message <> Reply to Message

Quote:And another minor improvement:
Allow to reorganize file tabs with drag and drop method.

Yeah, I really like "minor improvements" comprising 2000 lines

OK, puttin to ToDo. Do not expect it soon

Quote:#include "some_h_file.h"

Well, this is much easier. OK.

Subject: Re: Suggestions for improvements
Posted by mr_ped on Wed, 23 Nov 2005 00:21:29 GMT
View Forum Message <> Reply to Message

"Minor" in terms of usability impact.

I don't care if your IDE source needs refactoring to implement it, it's up to you to choose what's worth of the effort, and what will stay in ToDo for months/years.

I'm still testing the IDE during work-breaks, so here are some further things:

- I don't like the static context menu. It always shows only undo/redo/Copy'n'paste functions. Check MS Visual Studio (with or without Visual Assist), or vslick ... both do change context menu according to context under which it is requested. (that's the matter why it is called context menu) (I believe this one will be difficult too)

- when I use "Go to definition/declaration" in this case:

```
void foo(void) {
  //cursor is standing somewhere on the "foo2" text
  foo2();
}
```

it will take me to the foo() definition (i.e. 2 lines up), not to the definition of foo2(). This is quite different behaviour from vslick/vassist.
I already found out the correct way is to use Ctrl+W to query word, and than double click on the result, but this requires keyboard + mouse action, while common "go to ..." from vassist/vslick is usually single hotkey.

...
I'm testing the IDE on dumb pure C small code snippets, so I'm not sure how does my experience comply with large C++ project.

Subject: Re: Suggestions for improvements
Posted by mirek on Mon, 28 Nov 2005 09:14:33 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Tue, 22 November 2005 19:21
- I don't like the static context menu. It always shows only undo/redo/Copy'n'paste functions. Check MS Visual Studio (with or without Visual Assist), or vslick ... both do change context menu according to context under which it is requested. (that's the matter why it is called context menu) (I believe this one will be difficult too)

Actually, not really. The only problem is to figure out those contextes. Any hints how it should be changeing?

Subject: Re: Suggestions for improvements
Posted by gprentice on Mon, 28 Nov 2005 10:32:31 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 28 November 2005 04:14mr_ped wrote on Tue, 22 November 2005 19:21
- I don't like the static context menu. It always shows only undo/redo/Copy'n'paste functions. Check MS Visual Studio (with or without Visual Assist), or vslick ... both do change context menu according to context under which it is requested. (that's the matter why it is called context menu) (I believe this one will be difficult too)

Actually, not really. The only problem is to figure out those contextes. Any hints how it should be changeing?

I disagree with mr_ped comments. U++ changes the context menu according to which window the right click occurs on, just like other apps. I don't see any evidence of VS.NET 2003 changing the context menu when you right click in a source file. Slickedit has a long list of things in the context menu when you right click in a source file (plus nested menus too) and the only thing that changes is the "goto definition" and "goto reference" options - these specify the name of the identifier nearest to where the mouse click occurred. In slickedit, one of the options in the context menu is "edit this menu" which brings up a dialog allowing you to customise the menu (e.g. you can choose one of the many hundreds of "macros" to be fired up from a menu option).

If you want another suggestion for something really productivity enhancing is "completion" - type one or more characters and hit complete-previous or complete-next and you can step through the names that match that prefix. Just for fun, I was planning to look into whether this could be done using ESC sometime as I use this feature intensively.

Graeme

Subject: Re: Suggestions for improvements
Posted by mirek on Mon, 28 Nov 2005 11:52:56 GMT
View Forum Message <> Reply to Message

There two kinds of "completion"

- function(or type) name/signature completion, like function names etc..., just like after pressing -> or '.'. Of course, for global function names you need hotkey to activate this - the hotkey in TheIDE is Ctrl+Space. Means this is already there....

- automatic keyword completion - like after writing if(, TheIDE could offer ") {} else {}"

- invoked code completion - you time "ife" and press something, editor expands to "if() { } else { }".

Which one we are speaking about?

Mirek

Subject: Re: Suggestions for improvements
Posted by gprentice on Mon, 28 Nov 2005 12:21:41 GMT
View Forum Message <> Reply to Message

Quote:There two kinds of "completion"
Which one we are speaking about?

Um. None of the above.

Well, actually - the feature I was referring to was this

```
void somecode()
{
   int solid;
   so       // <<<<<  line X

   sort(a,b);
}
```

At line X, when I type so (or just s), I can now hit a hotkey for complete-previous. The editor first selects "solid" because that is the first identifier that matches the "so" prefix in the reverse direction. If I press the hotkey again it would select "somecode". If I change to the forwards direction it would go to solid and then sort. It highlights the identifier it chooses and writes it into line X. If you press any ordinary key (e.g. space) it assumes you accept the selected identifier and you carry on typing.

Actually the auto-list member names feature you already have is impressive and the other things you mention are also "highly desirable"

If I had to choose between auto expansion of "if space" or the completion thing I describe, I don't know which I'd choose. (They're all much more important than organizing file tabs with drag and drop though - an option for showing open files in the left pane (instead of files of highlighted package) would be more important than reorganize file tabs too! - in my opinion ...

Graeme

Subject: Re: Suggestions for improvements
Posted by mirek on Mon, 28 Nov 2005 15:05:27 GMT
View Forum Message <> Reply to Message

gprentice wrote on Mon, 28 November 2005 07:21

If I had to choose between auto expansion of "if space" or the completion thing I describe, I don't know which I'd choose. (They're all much more important than organizing file tabs with drag and drop though - an option for showing open files in the left pane (instead of files of highlighted package) would be more important than reorganize file tabs too! - in my opinion ...

But there are no open files in TheIDE There is just one file you are editing... Means only possible thing is some form of LRU list. And never forget to use Ctrl+Tab, that is the best option for dealing with LRU files AFAIK.

Anyway, I will consider adding <LRU> pseudopackage....

Subject: Re: Suggestions for improvements
Posted by gprentice on Mon, 28 Nov 2005 20:47:13 GMT
View Forum Message <> Reply to Message

Quote:
But there are no open files in TheIDE Smile There is just one file you are editing... Means only possible thing is some form of LRU list. And never forget to use Ctrl+Tab, that is the best option for dealing with LRU files AFAIK.

Anyway, I will consider adding <LRU> pseudopackage....

You've lost me just slightly. By open files, I was referring to the files that have a "file tab" - the ones that you can step through with Ctrl Alt left/right. I miss Ctrl shift tab for reversing ctrl tab stepping order.

I recall some discussion about this on the mailing list. I definitely don't want to tell you what to do or suggest priorities but I wouldn't be using slickedit for serious work if I hadn't been able to write my own "open files" list vertical toolbar. Most often I have this displaying "open buffers" (slickedit terminology) grouped into directories that can be collapsed (I don't use the collapsing much) but I can also have a single MRU list (most recently used). A feature I also consider essential is the ability to alternate between the last two files visited - e.g. suppose you use ctrl tab to step through files a,b,c,d,e - then alternating between last two files would go d,e,d,e,d,e indefinitely. I also have a "mode" for stepping backwards and forwards thru MRU list without altering the MRU list itself until you exit the mode - arrow keys step forwards/back - also numeric keys 1 to 9 "jump" to that position (but I don't use them a lot).

Anyway, LRU pseudo package sounds good but it would need to show only files that are "open" (that have a file tab), rather than show the entire history of files you have edited since time began

the way ctrl tab does - even better would be a hotkey or context menu option for switching between "all of time" and "current" files.

Graeme
